

STINFO COPY

**UNITED STATES AIR FORCE  
RESEARCH LABORATORY**

---

**The Role of Haptics in Service  
Manual Task Validation**

**Norman Badler  
Aaron Bloomfield**

**University of Pennsylvania  
Center for Human M&S  
200 South 33<sup>rd</sup> Street  
Philadelphia, PA 19104**

**Patrick J. Vincent**

**TASC, Incorporated  
2555 University Blvd.  
Fairborn, OH 45324**

**Kevin Abshire  
Lockheed Martin Corporation**

**Jeffrey L. Wampler  
Randy P. Allen  
Air Force Research Laboratory**

**February 2002**

**Final Report for the Period December 2000 to February 2002**

**20040218 113**

*Approved for public release; distribution is unlimited.*

**Human Effectiveness Directorate  
Deployment and Sustainment Division  
Sustainment Logistics Branch  
2698 G Street  
Wright-Patterson AFB OH 45433-7604**

## NOTICES

When US Government drawings, specifications or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications or other data, is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Please do not request copies of this report from the Air Force Research Laboratory. Additional copies may be purchased from:

National Technical Information Service  
5285 Port Royal Road  
Springfield, VA 22161

Federal Government agencies registered with the Defense Technical Information Center should direct requests for copies of this report to:

Defense Technical Information Center  
8725 John J. Kingman Rd., Ste 0944  
Ft. Belvoir, VA 22060-6218

## DISCLAIMER

This Technical Report is published as received and has not been edited by the Air Force Research Laboratory, Human Effectiveness Directorate.

## TECHNICAL REVIEW AND APPROVAL

AFRL-HE-WP-TR-2003-0140

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

//signed//  
MARK M. HOFFMAN  
Deputy Chief  
Deployment and Sustainment Division  
Air Force Research Laboratory

# REPORT DOCUMENTATION PAGE

Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE February 2002	3. REPORT TYPE AND DATES COVERED Final - December 2000 - February 2002
4. TITLE AND SUBTITLE The Role of Haptics in Service Manual Task Validation			5. FUNDING NUMBERS C: F33615-99-D-6001 DO: 08 PE: 62202F PR: 1710 TA: D0 WU: 09
6. AUTHOR(S) Norman Badler, Aaron Bloomfield, Patrick J. Vincent, Kevin Abshire, Jeffrey L. Wampler, Randy P. Allen			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of Pennsylvania Center for Human M&S 200 South 33rd Street Philadelphia, PA 19104 TASC, Incorporation 2555 University Blvd. Fairborn, OH 45324			8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory, Human Effectiveness Directorate Deployment and Sustainment Division Air Force Materiel Command Sustainment Logistics Branch Wright-Patterson AFB OH 45433-7604			10. SPONSORING/MONITORING AGENCY REPORT NUMBER  AFRL-HE-WP-TR-2003-0140
11. SUPPLEMENTARY NOTES			
12a. DISTRIBUTION AVAILABILITY STATEMENT  Approved for public release: distribution is unlimited.			12b. DISTRIBUTION CODE
13. ABSTRACT (Maximum 200 words) The primary objective of this research was to explore the role and feasibility of using haptics simulation for validating aircraft Service Maintenance Manual instructions. A task taxonomy was developed to facilitate understanding specific haptic simulation benefits, experiment designs, errors, and limitations. Alternative simulation approaches were investigated that might be used in combination with, substitution for, or augmentation of haptics to provide more realistic haptic action simulations. In general, haptic feedback for maintenance tasks requiring strength or constrained body configurations is strongly limited by present equipment capabilities. Manipulations requiring low force precision hand movements are somewhat better served by haptic feedback. A demonstration system explores haptic interaction with finger force feedback and hand force and torque feedback. Further research could be conducted to empirically test user abilities with haptic feedback against both non-haptic and actual physical manipulation. Recommendations may be used to guide or further focus efforts in the Service Manual Generation program and related efforts.			
14. SUBJECT TERMS Service Manuals    Maintenance Manuals    Technical Orders    Haptics Task Taxonomy    Technical Manual Validation    Technical Manual Verification			15. NUMBER OF PAGES 72
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT  UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE  UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT  UNCLASSIFIED	20. LIMITATION OF ABSTRACT  UL

**THIS PAGE INTENTIONALLY LEFT BLANK**



## Preface

This task was performed under Delivery Order #8 of the Technology for Readiness and Sustainment (TRS) contract (F33615-99-D-6001). The research was conducted from December 2000 through February 2002.

This effort is performing risk-reduction research in support of the Service Manual Generation (SMG) program. The SMG program is a dual-use research agreement with the Air Force Research Laboratory, General Electric and Lockheed Martin to automate critical aspects of the maintenance manual development process. The program is developing three enabling technologies: 1) exploded view generation techniques where Computer-aided Design (CAD) support tools are developed to create exploded views of part assemblies; 2) task generation where natural language techniques are applied to produce human-understandable maintenance task descriptions and; 3) a haptic-enabled virtual reality (VR) validation environment which allows maintenance task rehearsal to reveal inconsistencies, errors, and other potential problems with the generated maintenance task descriptions. The VR environment will also serve as a revolutionary medium for maintenance training.

Haptic feedback simulating force feedback may be the critical link in this VR simulation technology. The purpose of this task was to understand, evaluate, and establish benefits and limitations of this approach. In addition, alternative simulation approaches were investigated that might be used in combination with, substitution for, or augmentation of haptics to provide more realistic haptic action simulations. Our recommendations may be used to guide or further focus efforts in the SMG program and related efforts.

In general, haptic feedback for maintenance tasks requiring strength or constrained body configurations is strongly limited by present equipment capabilities. Manipulations requiring low force precision hand movements are somewhat better served by haptic feedback. A demonstration system explores haptic interaction with finger force feedback and hand force and torque feedback. Further research should be conducted to empirically test user abilities with haptic feedback against both non-haptic and actual physical manipulation.

# Table of Contents

	Table of Contents .....	iv
	List of Figures & Tables .....	vi
I.	Introduction.....	1
II.	Virtual Reality Research Review.....	1
	A.    Ideal Haptics Configurations .....	3
	B.    Current Haptics Technology.....	3
	C.    Current Research.....	5
	D.    Alternative Approaches .....	8
	E.    Differences Between Haptics Environments .....	8
III.	Aircraft Maintenance Action Taxonomy Development .....	11
	A.    Haptic device configurations .....	11
	B.    Force Types.....	11
	C.    Action Types.....	12
	D.    Task Taxonomy .....	14
IV.	CAD Geometry Development.....	17
V.	Assessment of Virtual Maintenance Actions.....	18
	A.    Effectiveness of Haptics Technology to Simulate Tasks.....	18
	B.    Testing Requirements .....	27
	C.    Technical Hurdles .....	30
	D.    Benchmarks and Comparisons Among Haptics Platforms.....	31
	E.    Evaluation Results .....	32
VI.	Failure Condition Analysis .....	33
	A.    Error Types .....	33
	B.    Error List.....	35
VII.	Validation Methodology Development.....	36
	A.    Simulation Development and Setup.....	36
	B.    Simulation Execution.....	39
	C.    Evaluation Design.....	40
	D.    Analysis.....	41

VIII.	Demonstrate the Assessment & Validation Approach.....	45
A.	Simulation Development and Setup.....	45
B.	Simulation Execution.....	48
C.	Evaluation Design.....	49
D.	Analysis.....	49
IX.	Conclusion .....	54
X.	Appendix A.....	57
A.	Modeling the Bayonet Connector .....	57
XI.	References.....	61

## List of Figures & Tables

Figure 1: Phantom robotic arm .....	4
Figure 2: CyberGlove .....	5
Figure 3: CyberGrasp.....	5
Figure 4: Bayonet connector simulation.....	49
Figure 5: Bayonet connector: the tube .....	57
Figure 6: Bayonet connector: the insertion groove.....	58
Figure 7: Bayonet connector: the rotation groove .....	58
Figure 8: Bayonet connector: the lock groove.....	59
Figure 9: Bayonet connector: completed transparency view.....	59
Figure 10: Bayonet connector: completed wireframe view.....	60
Figure 11: Key for the bayonet connector .....	60

Table 1: Differences in Virtual and Haptic Environments .....	10
Table 2: Aircraft Maintenance Task Taxonomy .....	15
Table 3: Taxonomy Task Simulation Effectiveness Summary.....	27
Table 4: Task Taxonomy Error List.....	35

## I. Introduction

An effective Virtual Reality (VR) experience consists of computer simulations that stimulate human sensory inputs. Two of the primary sensory channels are vision and kinesthetics. The visual channel presents imagery of a 3D environment and user operations or head movements may be used to navigate the visual space. The kinesthetic channel presents feelings of solidity, contact, pressure, or force. This study focuses on the issues underlying the use of Virtual Reality haptic technology for validating tasks in Service Manuals. The Service Manual Generation (SMG) Dual Use effort is combining solid modeling software with VR and force-feedback devices to create a simulation platform for maintenance task analysis. The concept behind SMG is to allow the analyst to virtually perform the specified task and assess whether it makes sense, is complete, and takes into account safety, human factors, and related performance problems.

In support of this endeavor, the AFRL/HESS has invested in series of research tasks over the last several years that have focused on specific technologies that may be considered elements of a unified solution set for the simulation of aircraft maintenance procedures. The research tasks include substantial work in automating procedural language (contract F41624-97-D-5002, D.O. 8); the analysis of Automating Maintenance Instructions (AMI)-related data in Computer-Aided Design (CAD) and Computer-Aided Manufacturing (CAM) models (contract F41624-97-D-5002, D.O. 14); the use of Parameterized Action Representations (PARs) to describe maintenance actions in a format suitable for human modeling and simulation (contract F41624-97-D-5002, D.O. 17); a report on Technical Orders entitled Design Concepts for Automating Maintenance Instructions (contract F33615-99-D-6001); and, most importantly, a report titled Technology for Maintenance Procedure Validation (contract F33615-99-D-6001).

The goal of this study is to investigate virtual task validation using haptics. For generality we examined several validation approaches:

1. Interactive user task attempts and analysis
  - a. Using visual and haptic feedback
  - b. Using visual feedback only
2. Non-interactive task attempts and analysis

Case 1b provides an alternative to haptics and must be explored even if only as experimental controls. In case 2, "pure" computation would be used to establish task validity. In any of these cases, task validation means that one can establish that a given maintenance task could or could not be performed. There are four possible general outcomes:

- A. The task is physically possible and humanly possible by any "typical" aircraft maintainer.
- B. The task is physically possible but unreasonable to expect from a "typical" aircraft maintainer (e.g., insufficient strength).
- C. The task is physically impossible due to human limitations (of any maintainer).
- D. The task is physically impossible due to physical limitations (part is just inaccessible or not extractable).

Clearly this problem is complicated by the need to characterize aircraft maintainers and their statistical capabilities with respect to anthropometry, dexterity, strength, and skill. Our haptic study does not overtly address some of these variables though they are critically important to assess situations A, B, and C. Human factors professionals will typically use several subjects to assess task validity for cases A and B. Existing sources for such data include the *Human Factors Design Handbook* [WTT92].

This document is organized generally as follows. First, we review and evaluate Virtual Reality haptics interfaces to develop a base for interactive haptics simulations. Then we examine a taxonomy of maintenance tasks and their haptic simulation requirements, characteristics, and errors. We explore one interesting task type within the taxonomy (because it includes force and torque components simultaneously) by studying the haptic simulation of a bayonet connector insertion. We then evaluate the likely effectiveness of haptics simulations relative to all task taxonomy types and force/torque requirements. Non-haptic interactive and purely computational approaches to task validation are considered. User errors and system technical limitations are considered. The demonstration using the bayonet connector model is discussed and evaluated. Finally we consider further validation methodology development and open issues.

## II. Virtual Reality Research Review

*Goals: To perform a review of current haptics and related VR technology research that may be applicable to service manual validation and recommend alternative approaches to the proposed SMG configurations to augment the haptic technology for validation purposes.*

### A. Ideal Haptics Configurations

Essentially, haptics interaction technology is far from ideal. The ultimate solution would be a system that could completely deceive an individual that he or she was in a virtual world. Examples of these only exist in science fiction – the Holodeck of Star Trek (holograms and various solid objects create the illusion of the virtual world) or the neurological connection of the movie “The Matrix” (the computer system taps directly into one’s neurological system, thus completely deceiving the individual. Well, almost completely, as the movie shows). Although these systems will not be obtainable in the near future (if ever), they are useful for comparison purposes, as they can be called the “perfect” haptic interaction – what every other simulation should strive for.

### B. Current Haptics Technology

Enter any computer store, and haptic devices surround you. There are joysticks that can exert some force on the user’s hand; mice and trackballs that can apply resistance when you move them; devices for the blind, who obviously can’t use visual stimuli. While these devices are certainly within the realm of haptic devices, they are not the study of serious haptic research. Thus, this section’s overview of current haptic technology is limited to high-end haptic devices.

One extreme configuration possible with today’s technology (if money and safety were not problems) would be a fully robotic exoskeleton. There has been some work in this area, but the primary purpose of the exoskeleton is often non-haptic. One example is the GE “Hardiman” exoskeleton (built in the 1960’s and 1970’s) which was used to increase the end-effector strength of the human operator [Bur96]. Sarcos Research also built a similar but more dexterous device for arm strength enhancement in the early 1990’s.

A full haptic exoskeleton would allow for rigid collisions. Currently, when something collides with an object, there is often a limited amount of force that can be applied to the user. Sometimes no resistive force (such as in the case of the actuators, discussed below) can be applied at all. An exoskeleton would enable resistive forces to be applied to any part of the body.

There are significant hurdles to constructing such a haptic environment. The first and foremost is the monetary cost. In addition to building a human-sized robotic exoskeleton, a significant amount of computational power would be required to run it, and thus a lot of program code and programmers. Another concern is safety. The GE exoskeleton was terminated for a number of reasons, one being user safety during a hydraulic leak. The haptic benefit of an

exoskeleton is not firmly established (hence the animus of this research task). Thus, the benefit-to-cost ratio is not fully known, and because of the cost, this is not currently feasible. It is believed that this will be the future of haptic environments.

There are a number of haptic products out today. One category is a robotic arm that can exert force and torque upon the user holding the end of the arm. This is what the Phantom device (shown in figure 1) does, although many other similar haptic products exist. By pushing back on the user's hand, the robotic arm can simulate the feel of any sort of object – from a basic plane to a kidney shaped bean. The disadvantage to these devices is that your haptic action must be within the range of the length of the robotic arm [Phan01]. These seem to be the most popular haptic device today.



Figure 1: Phantom robotic arm

Courtesy of [Phan01]

Another class of haptic products includes CyberGlove and CyberGrasp. CyberGlove, shown in figure 2 [Cyg101], is a glove with 18 angular sensors. These sensors allow a computer system to record the angular displacements of each of the joints in the hands (the four fingers each have three knuckles, the thumb has two, and in-between the fingers and knuckles). Thus, an exact simulation of the state of the user's hand can be obtained, after a few measurements of the user's hand size. CyberGrasp, shown in figure 3 [Cybr01], is a device that allows force to be applied to the fingers by pulling them back. Essentially, it has a pulley system that is attached to the back of the hand, and each finger is attached to a string that can tighten, thus pulling the fingers back. These devices have a greater range than the Phantom.





Figure 2: CyberGlove

Courtesy of [Cyg101]



Figure 3: CyberGrasp

Courtesy of [Cygr01]

Recently, there have been efforts to combine the two previous devices. Immersion Corporation, the company that manufactures CyberGlove and CyberGrasp, has a new device (called CyberForce) that has a robotic arm that can exert force on a hand that is already in a CyberGlove and CyberGrasp [Cybf01].

There have been other examples of haptic devices that apply forces to the individual fingers. The Dextrous Master is a box-shaped device that has string attached to the fingers. By tightening the individual strings, forces are applied to the individual fingers. This device, however, has an even more limited range than the Phantom, as the hand must keep in the box [Bur96].

In summary, current haptic devices are in their infancy. There is a small range of devices, each being applicable to a range of tasks. The devices continue to develop and improve in their capabilities while lower cost haptic devices are starting to trickle down to the average consumer.

## C. Current Research

Two areas of research are presented here. The first, and the older of the two, is the research into the various taxonomies of hand grasps. The second area is current haptics research.

### 1. Taxonomy Research

There has been a fair amount of research in the area of categorizing the grasps of the human hand. Venkataraman and Iberall, in their 1988 book Dextrous Robot Hands, provide a summary of grasps of the human hand, the focus of their book. This summary is reproduced here [VI88].

One of the earliest grasp taxonomies was first defined by Sclesinger in 1919, and later summarized by Taylor in 1955. They define six grasps of the hand: cylindrical, fingertip, hook, palmar, spherical, and lateral. The object being grasped determines the choice of grasp – so if you are grabbing a sphere, you use a spherical grasp [Sc19,TS55].

Napier in 1956 suggested that you should categorize by function rather than appearance (the reason was that when you are opening a jar, you are first using a power grip to get it unstuck, then a dexterous grip to remove the lid). He defines power grasps, when strength is needed, versus precision grasps, when dexterity is needed [Nap56].

Arbib in 1985 discussed virtual fingers - when picking up a pencil, the number of fingers opposing a thumb doesn't really matter (as long as it's greater than zero), so they can be grouped as a single "virtual finger". The thumb forms the other virtual finger [AIL85].

Iberall in 1987 described grasping in terms of "oppositions", which defines three such oppositions: pad, for forces between the pads of the fingers and thumb; palm, for forces between fingers and the palm; and side, for forces between the thumb and the side of the index finger. These oppositions can be done separately or simultaneously. Each task uses two virtual fingers, one of which is the thumb or palm [Ibe87].

Cutkosky in 1989 defined a hierarchical tree structure taxonomy for grasps. This tree, while complete for maintenance tasks (the focus of the article), is not exhaustive of all hand grasps - for example, holding a cigarette between the index and middle fingers is not provided by this model [Cut89].

The focus of both of these taxonomies focus on how a hand grasps an object, and not necessarily the hand/arm action required (which is our focus in the next section). These taxonomies also were not designed with haptics, and specifically haptic devices, in mind. There has been very little computer related research involving the actions and motions of the hands and arms together.

## 2. Haptics Research

Most of the research in the haptics field today focuses on how to better use the existing devices that are currently commercially available. Much of the research is done on the Phantom, as it provides a high degree of accuracy and is usable for a large variety of tasks.

All haptic actions boil down to a few basic movements. Miller and Zeleznik describe a series of these basic movements, including push buttons, grooves, ridges, and notches [MZ99]. Their work is particularly useful, as it forms the basis for anyone developing haptic environments.

A large area of haptic research is the area of assisting the visually impaired. The sense of touch is vital to the visually impaired, as they have one less sense (sight) that they can effectively use. Hence, the use of Braille. Colwell et al., studies the usefulness of haptics as a means for visually impaired people to recognize textures and objects [CPK+98]. Kurze describes a system, based on an analysis of drawing techniques used by visually impaired people, for using haptics for drawing real world objects [Kur97]. Ramloll et al., attempts to make line graphs, which are so easy to view by those with sight, accessible to visually impaired people through auditory and

haptic stimulus [RYB+00]. Ramstein uses a haptic Braille system to try to assist the visually impaired [Ram96]. The system isn't as good as regular Braille, but it is showing improvements. The Pantograph, also by Ramstein, is a haptic system especially designed for use by visually impaired people in an office setting [RH94]. This has been expanded to synchronize auditory cues along with the haptic sensations [DP00].

A particularly interesting, and potentially one of the most useful, applications of haptics is the integration with the computer desktop. Miller and Zeleznik add haptics to X Windows: "additions include adding ridges around icons and menu items to aid interaction, alignment guides for moving windows, and other enhancements to window manipulation" [MZ98]. Munch and Dillmann extend that concept to try to predict which widget a custom-built haptic enabled mouse is moving towards. The idea is to stop the mouse when it enters that widget [MD97]. Although these both dealt with X Windows running on a UNIX system, the concepts could easily be applied to a Microsoft Windows system.

As with all computer graphic environments, the complexity of the scene can rapidly lead to deteriorating performance. Ruspini et al., and Gregory et al., describe heuristics for dealing with complex graphical environments [RKK97,GME+00]. Both systems use the Phantom robotic arm, which requires update rates of 1000 Hz. Using methods such as geometric locality and temporal coherence, they were able to interact with much more complex scenes.

A number of researchers have focused on studies on the usefulness of haptics, rather than solely on developing new systems. Wang and MacKenzie explored how useful haptic and virtual reality contextual information was compared to a real-world situation. They used tables, and compared how hard it is to manipulate something on a real table versus manipulating something on a virtual one [WM00]. Sallnas et al., show that "haptic force feedback significantly improves task performance, perceived task performance, and perceived virtual presence in the collaborative distributed environment". They used a collaborative desktop virtual environment for their study.

The majority of models used in haptic environments are geometry based, using either pure geometry models or polygonal representations of such. Avila and Sobierajski and McNeely et al., both discuss ways to have haptic feedback use voxel sets [AS96,MPT99]. Both use Phantom robotic arms, which (as mentioned above) require a 1000 Hz update rate.

Lastly, there are a number of other miscellaneous areas. Thompson et al., describe a system that links a Phantom-like robotic arm with a CAD modeling system to allow sculpting of models [TJC97]. Noma et al., uses a 3D mouse for control of remote objects [NMK96]. Lawrence et al., describes the use of haptics "to allow exploration and understanding of fluid dynamics data" [LLPN00]. Dachille uses a Phantom device to allow the creation and deformation, via a mass-spring setup of a B-spline surface [DQKES99]. Jack et al., use a haptic system to aid with the rehabilitation of the hands of stroke victims. They use a CyberGlove for positional data and another glove for force feedback [JBM+00].

There are numerous other papers on haptics. One search (at <http://liinwww.ira.uka.de/bibliography/index.html>) yielded over 200 papers, books, and articles with just the keyword "haptic". Other haptic related papers might not include that word, but would include phrases such as force feedback. Examining all of the 200 plus papers is beyond the scope of this report. We believe that the summary presented here is indicative of the overall research that currently exists in the area of haptics

### 3. Maintenance and Validation Research

We found very few papers on the subject of maintenance validation using haptics. While there are many reports on maintenance validation (most notably, previous Air Force work orders) and many reports on haptics, very little combines the two fields.

A previous DO8 report, Technology for Maintenance Procedure Validation, discussed several automated Technical Order validation system components. The result of that study was that a number of areas of research are still needed, including Parameterized Action Representations (PARs), translators from PARs to Technical Orders, and the use of disassembly planners to assure spatial access. Some of this research was completed in prior research work efforts.

There is increasing interest in validating task simulations against human actions [Cha01]. Unfortunately, even this recent survey fails to give a general task framework or experimental methodology for haptic validations. Haptics simulation has focused on medical applications such as surgical training and tumor detection [WZM+01, Isd01], with only a few efforts in manufacturing, assembly, or repair [JJW+99, AKH01].

#### D. Alternative Approaches

There are some other approaches to haptic simulations. The products described above focus on providing sensation to the hand. In the real world, we obtain haptic stimulus from our entire body. One promising approach is with cutaneous actuators (tactors), which are coin-sized devices that can exert a small, localized pressure when activated. By putting them on the skin, their expansion can exert a sensation on the recipient. The Navy is experimenting with such devices mounted on a flight jacket to give a pilot cutaneous sensation of the true gravity vector or of targeting opportunities. For manual tasks, one could construct a suit, or even just a sleeve for one arm, that contained a number of these devices. If the arm collided with an object in the virtual world, one or more actuators would be activated, providing instant haptic feedback about the collision. This would not provide any resistive force against the arm, but it would allow the wearer to immediately actualize the collision with an object. This may be significant for virtual aircraft maintenance, as one may then be able to sense and thus maneuver around obstacles to perform a desired task.

#### E. Differences Between Haptics Environments

We have compiled a list of the similarities between our virtual environment and GE's proposed virtual environment for the SMG effort, in Table 1. The last column lists the issues that pertain to porting our work to GE's platform. The only significant difference is in the third haptic device (CyberGrasp), which GE does not have (they do have a CyberGlove). Thus, any simulation developed with a CyberGrasp would not be able to be run at GE's research lab unless one was procured.

	UPenn's configuration	GE's configuration	Portability Issues
Visor / HMD	Virtual Technologies V8 <ul style="list-style-type: none"> <li>VGA (640x480) resolution</li> </ul>	Kaiser XL50 <ul style="list-style-type: none"> <li>XGA (1024x768) resolution</li> </ul>	Minimal. Both devices accept 15-pin video inputs, and the only change required is to change the output resolution
Data Glove	Virtual Technologies CyberGlove	Virtual Technologies CyberGlove	None. Devices are identical.
Head tracking	Ascension Inertial Tracker	Ascension Flock of Birds	Minimal. Both of the trackers' output must be converted to (x, y, z) coordinates for the graphics library, and thus the only difference is the single method to do the conversion. For the inertial tracker, that consists of three simple formulae.
Hand tracking	Ascension Flock of Birds	Ascension Flock of Birds	None. Devices are identical.
Haptic device 1	SensAble 3.0 (a.k.a. Phantom) with 6 degrees of freedom	SensAble 1.5 (a.k.a. Phantom) with 6 degrees of freedom	None. Devices are identical.
Haptic device 2	SensAble 3.0 (a.k.a. Phantom) with 3 degrees of freedom	SensAble 3.0 (a.k.a. Phantom) with 3 degrees of freedom	None. Devices are identical.
Haptic device 3	Virtual Technologies CyberGrasp	(none)	Significant. Since GE does not have a CyberGrasp, this may cause a problem, as the Phantoms cannot replicate the haptic sensations that the CyberGrasp can.
Operating System	Windows NT 4.0	Windows 2000	Minimal. The programs developed on the NT 4.0 platform are tested on a Windows 2000 platform.
Processor	Dual Pentium III, 850 MHz	Dual Pentium III, 866 MHz	Minimal, as the speeds are practically the same.
Memory	256 Mb	512 Mb	Minimal, as our programs will not use greater than 256 Mb.
Video Card	Intergraph Intense 3D Pro	Elsa Synergy III	Unknown. However, the only effect would be on graphics performance,

	3D Pro		which should not be adversely affected, as good graphic cards are inexpensive.
--	--------	--	--

**Table 1: Differences in Virtual and Haptic Environments**

### III. Aircraft Maintenance Action Taxonomy Development

*Goals: To develop a taxonomy of typical aircraft maintenance actions, representative of the entire spectrum of possible actions and to define practical benchmarks for an SMG virtual validation environment.*

In this section we present a task taxonomy that we believe includes the full range of aircraft maintenance actions. The testing requirements are included in Section 3.5.4 (the Assessment of Virtual Maintenance Actions in the Testing Requirements Section), as it fits better into that section. Many of the aspects of the taxonomy presented in this Section are described later in the report, such as the error types, validation methods, and testing requirements.

#### A. Haptic device configurations

The following is a partial list of possible haptic configurations available today. These are the possible platforms available with a Phantom, CyberGrasp, and a CyberGlove. As they are commercially available devices, we used them to form the basis for our task taxonomy.

1. No devices for either hand
2. Devices for one hand only. Total of 4 configurations.
  - a) No devices (hands free)
  - b) CyberGrasp
  - c) Phantom
  - d) CyberGrasp and Phantom
3. For both hands any of the above four combinations ("No devices" is replaced by a rigid support grasping bar) on one hand with any of the above combinations for the other hand. Total of 16 configurations. The configuration with both hands on rigid grasping bars is not relevant, making 15 configurations to be considered.
4. A full exoskeleton with haptic sensors and actuators

#### B. Force Types

There are a three force categories and three torque categories in our task taxonomy. These are explained below:

- "No Force" is when no force is needed to perform the action.



- "Force Only I" is a force direction aligned with the motion of the hand or arm, such as pushing a door.
- "Force Only II" is a force direction not aligned with the motion of the hand or arm, such as sanding a block of wood.
- "No Torque" is when no torque is needed to perform the action.
- "Torque Only I" is a torque axis through grip space (the axis of the torque is aligned with the hand or arm), such as using a screwdriver.
- "Torque Only II" is a torque axis that is not through grip space (the axis of the torque is offset from the hand and arm), such as using a lever device (i.e. a wrench) for leverage.

Combining those together, we obtain a total of nine force/torque combinations. The combination of no force and no torque is not relevant to haptics research, so it is ignored. The two cases of only force and no torque, or only torque and no force, are listed in the taxonomy table that follows. That leaves four categories of the combination of both force and torque. We have decided to combine them into one category, as the distinctions of these four combinations are blurred.

### C. Action Types

The task taxonomy contains eight categories of actions. Note that these actions focus primarily on the type of movements that the hands perform. To a lesser extent, the actions that the arms perform are a part of it, because they position the hands in space. This is attributed to aircraft maintenance actions being performed primarily with the hands, and current haptic devices are designed for the hands.

Thus, there are a lot of actions that are not included in this taxonomy. Running and walking, for example, require no use of the hands and arms. While this may be a subject of future research, it is beyond the scope of this report.

The categories are chosen based on a high-level view of the type of task being performed. This is not necessarily the type of motion being performed. For example, a tactile pressure task includes such diverse actions as dialing a rotary telephone, pushing a button, and loading grease into a hole.

The eight categories are described below. A brief description includes the ease of simulating each task; more detail on that subject is in section 3.5.4 (assessment of virtual maintenance actions).

#### **Fine motor control**

Fine motor control tasks are actions that require very fine guidance by the hands, and very precise movements. Example actions include pushing a pin into a hole (the alignment of the head of the pin must be right at the hole) or turning a dial (turning the dial to a specific



location requires a lot of fine motor control). These tasks require haptic devices that have a lot of precision (such as the Phantom), and are a natural choice for simulation

### **Significant arm strength**

This category of tasks includes any sort of task that requires a significant amount of strength to perform. Note that if you take out the strength component, this task may fall under another category. Example actions include pulling open a stuck access panel and turning a stuck valve. These tasks are often difficult to simulate, as many haptic devices cannot exert the necessary force to make the action realistic.

### **Tactile (finger pressure) friction**

These tasks are similar to the fine motor control tasks. The difference is that these tasks rely more heavily on haptic feedback. If one were to have no feeling in one's arm, one could still insert a pin into a hole (a fine motor task) by using eye-hand coordination. Other tasks, however, cannot be properly performed by just using eye-hand coordination. Pushing a button, for example, requires the feeling of the button "releasing" the pressure as it is pressed. While one could perform this action without haptics, the idea is that one knows one has completed the action when they feel the release of the button's pressure (as opposed to seeing the pin inserted all the way into the hole). These tasks, like the fine motor control tasks, are good choices for simulation.

### **Cooperative two-handed tasks**

Most tasks that require two hands to perform fall under this category. The one exception is where one hand is bracing against something (that falls under the next category). Note that each hand may be performing another task type – most often (but not always), a significant arm strength task. Examples include lifting a heavy load or dumping a wheelbarrow. These are often difficult to simulate, as they require multiple haptic devices and thus multiple computers to create the virtual and haptic environment.

### **Braced two-handed tasks**

Any two-handed task where one hand is braced (against a wall, a bar, etc.) falls under this category. As with the previous category (cooperative two-handed tasks), the one hand that is not being braced is often performing a significant arm strength task. These are easier to simulate than the cooperative tasks, as you do not need a haptic device on the hand that is used as the brace.

Note that there are other tasks that can use a brace, but do not require significant arm strength. Examples of these would be bracing for balance, or bracing to hold a light object steady. These would all fall under the cooperative two-handed task category. The tasks in the braced category are those tasks that require bracing because the action requires a lot of arm strength.

### Manipulating a deformable object

Deformable objects are difficult to model via a computer, and are the focus of a lot of current research. Often times an action being performed is a deformable object. Examples include wringing out a towel, pushing a wire-like connector into a socket, or holding a cloth steady in the wind. These tasks require complicated computer models to allow for realistic deformations. Note that having the Phantom prod a deformable object is considered a tool-assisted task. This category covers interactions with the deformable object directly.

### Tool-assisted tasks

Any task that requires a hand tool to perform falls under this category. Such tools include hammers, screwdrivers, wrenches, crowbars, etc. These tasks are good choices for simulation, as the shaft of the Phantom can easily simulate the handle of the tool being used in the virtual environment.

### Multi-finger tasks

These tasks include anything that requires more than one finger to perform. The use of the Phantom is mainly by holding a steel shaft. The CyberGrasp allows for individual finger forces. A non-maintenance example includes playing a musical instrument. Aircraft maintenance examples include turning a dial (multiple fingers grip the sides of the dial) and pulling a pin by the head (multiple fingers grip the head of the pin).

## D. Task Taxonomy

The task taxonomy is depicted in Table 2. The bolded entries are explained below the table.

	Force Only I	Force Only II	Torque Only I	Torque Only II	Force & Torque
Fine motor control	<b>Pushing a pin</b>	Wiping off grease	Turning a dial	<b>Using an X-wrench</b>	<b>Inserting a bayonet connector</b>
Significant arm strength	Pulling open a stuck access panel	Sanding	Turning a stuck valve	Using a wrench; turning a crank	Pushing a heavy door while turning a lever latch
Tactile (finger pressure) friction	Pushing a button	Loading grease into a hole	<b>Inserting a small bolt</b>	Dialing a rotary-type phone	<b>Inserting a bayonet connector</b>

Cooperative two-handed tasks	Lifting a bulky object; Pushing two connectors together	Filing with a large file	Extracting a large threaded rod or bolt	Dumping a wheelbarrow load	Pulling and twisting a piston from a cylinder
Braced two-handed tasks	Holding a support while doing an arm strength task	Holding a support while doing an arm strength task	Holding a support while doing an arm strength task	Holding a support while doing an arm strength task	Holding a support while doing an arm strength task
Manipulating a deformable object	Pushing to create a shape (fuel bladder removal)	Holding a cloth steady as it flaps in the wind	Wringing a towel	Stirring a viscous liquid	Twisting wires while pulling the cable taut
Tool-assisted tasks	Interface to increase force per unit area (hammer, chisel)	Interface to overcome friction or to increase force per unit area (plane, crowbar)	<b>Interface to increase torque (hex screwdriver)</b>	<b>Interface to increase torque via leverage (wrench)</b>	Interface to increase torque and force per unit area (screwdriver)
Multi-finger tasks	Pushing multiple buttons at once	Pulling a pin by its head	Turning a dial	Turning a large wing nut	<b>Inserting a bayonet connector</b>

Table 2: Aircraft Maintenance Task Taxonomy

After consultation with members of the SMG team, six sample actions were identified for prototyping and evaluation. These are shown in bold in Table 2. Note that there are more than six bold cells in the taxonomy table because one action (inserting a bayonet connector) appears multiple times in the table.

1. Fine motor control / Force Only I: Pushing a pin
2. Fine motor control / Torque Only II: Using an X-wrench
3. Fine motor control / Force & Torque, Tactile / Force & Torque: Inserting a bayonet connector
4. Tactile / Torque Only I: Inserting a small bolt.
5. Tool-assisted / Torque Only I: Interface to increase torque (hex screwdriver)
6. Tool-assisted / Torque Only II: Interface to increase torque via leverage (wrench)

The descriptions of the force and torque types appear above the taxonomy table. Note that it was also suggested to evaluate the task in the Cooperative two-handed / Force Only I cell (pushing two connectors together or lifting a bulky object), but since we only have one Phantom device, this could not be evaluated.

## IV. CAD Geometry Development

*Goal: To develop the CAD geometry required to simulate each of the maintenance tasks developed in the taxonomy in a haptics enabled virtual simulation environment.*

The demonstration programs involved individual haptic actions, which required smaller and simpler computer models. This allowed for greater graphical realism, as we were able to add lighting and textures without detracting from the overall system performance.

The majority of the geometry development consisted of the bayonet connector model. Other haptic simulations (pushing a button, turning a wrench) did not require significant geometry development.

Recall that a bayonet connector operates like a medicine bottle - you must push in the key (the top of the medicine bottle) while resistive force is applied by the bayonet connector, rotate the key, and then release the key, which will lock or unlock the bayonet connector, depending on which direction you are going. In our bayonet model, we used the following nomenclature.

We have modeled our bayonet connector using Maya, a 3-D modeling and animation program. We developed both images and an animation to show how our model works. Rather than include all of the images and the animation in this report, we have included only a few, and put all the images (and the animation) online at <http://hms.upenn.edu/software/AF/haptics/bayonet/>. To view the page, log in as guest with password a1b2c3d4.

Note that the Phantom does not read Maya files. The majority of the modeling of the bayonet connector was done by writing C++ code, so that the Phantom could properly interact with the model. This code consisted of constructive geometry primitives. Constructive solid geometry uses basic primitive shapes (spheres, boxes, cones, cylinders, etc.) and Boolean operations (union, intersection, and difference) to construct more complicated geometrical representations. The only shapes we needed for our model were the infinite cylinder (which has no end caps - it goes on forever, hence its name) and the box.

A pictorial description of the modeling of a bayonet connector is in Appendix A.

## V. Assessment of Virtual Maintenance Actions

*Goal: To perform an assessment of the effectiveness of haptics and related VR technologies to validate each action in the taxonomy.*

A fair amount of the work required for this section appears in other places in this report. The evaluations that were designed for each task/force combination are the specific task examples shown in the task taxonomy table. That table also identifies the potential failure conditions, which are also discussed in the next section (3.5.5, Failure Condition Analysis).

The assessment that follows is based on how easily and how realistically the actions can be simulated with today's haptics technology. As technology changes, these tasks will have to be re-evaluated. However, the criteria for evaluating them will not change.

### A. Effectiveness of Haptics Technology to Simulate Tasks

For each of the tasks in the taxonomy, we have assessed their effectiveness for simulation using current haptics technology and virtual environments. The specific haptic devices we used in our evaluation, and the description below, were the Phantom and the CyberGrasp. The results appear below. Each task/force combination was given a rating of poor, moderate, good, or excellent, depending on how realistic a simulation could be developed with current technology. Most of the tasks have the same rating for all the force/torque combinations. The two exceptions are multi-finger tasks and cooperative two-handed tasks.

#### 1. Criteria

We have defined the criteria as follows to determine how effective current haptics technology can be to simulate a particular task type.

For each taxon, a 0, 1, or 2 (or, in the case of the first criteria, a 0 through 4) will be assigned. A 0 means the taxon does not fulfill the requirement at all, and a 2 (or 4, in the case of the first criteria) means it does fulfill the requirement. In some cases, non-integer values were assigned (1.5, for example). The sum of these ratings will be a number from 0 to 10, which will reflect how well an action can be simulated by current haptics technology.

1. *Application of force & torque:* The haptic device can simulate the needed forces and torques in the correct direction(s) needed for the task simulation. This criterion is more important than the others, so it is rated out of a possible 4 points (whereas the others are rated out of 2).

2. *Sufficient force & torque*: The haptic device can provide sufficient force and/or torque in the right direction to realistically simulate the haptic action. The previous criteria determined whether it could provide the forces and torques in the correct directions. This one is determined by whether the device can provide enough force and torque for a realistic simulation. Note that if the device cannot provide the force in the right direction (from the previous criterion), then it cannot provide sufficient force and torque.
3. *Grasp simulation*: Haptic device can simulate the feel of grasping the object in the simulation. For example, the shaft of the Phantom can simulate the holding of a handle of a tool very realistically, but it cannot simulate the holding of a deformable object (such as a piece of cloth). It can simulate interaction with said cloth, but that means it is providing the right forces and torques.
4. *Range of motion*: The haptic device allows for the proper range of motion that the haptic task requires. The Phantom, in particular, is not very mobile, which may be needed for some tasks. Some haptic simulations will have to be scaled down if the range of motion of the haptic device is not enough.

The rating scale, found below, is determined based on how much the particular task can fulfill these requirements with current haptics technology. Some of the requirements will be easy to determine – the Phantom can simulate any and all forces and torques needed, but cannot simulate the feel of holding a deformable object. Some of the requirements will be somewhat more vague as to how a particular task fulfills it. For example, the Phantom does not allow a huge range of motion, but it can be enough if the task does not require very much range of motion to be performed.

- Excellent: A rating of 9 or 10
- Good: A rating of 7 or 8
- Moderate: A rating of 5 or 6
- Poor: A rating of 2, 3, or 4
- N/A: A rating of 0 or 1

In the future, experimentation may show us that we need to modify the relative weights of the criteria, or to change the ratings scale.

## 2. Task Breakdown

We considered two haptic devices for our evaluation, the Phantom and the CyberGrasp. While other devices exist, these are by far the most common. For each of the breakdowns below, we detail how we rated each action for both of the devices. Only the device that received the higher rating is considered when comparing the tasks. Unless otherwise noted, all the assessments apply to all the force and torque categories.

### **Fine motor control tasks**

These tasks can be simulated very realistically.

#### **Phantom:**

- Application of force and torque: **4**. The Phantom is capable of very minute movements and can apply very small amounts of force and torque. It also has a high amount of precision for determining where the shaft is, unlike other haptic devices that require electromagnetic trackers to determine their position and orientation. The Phantom runs at 1000 Hz, which means it is able to adapt very rapidly to any movements, and can provide all 6 degrees of freedom of forces and torques.
- Sufficient force and torque: **2**. Fine motor control tasks do not require a large amount of force or torque, which the Phantom can provide.
- Grasp simulation: **1**. The only drawback is that the Phantom can only provide the grasp feeling of holding a shaft. Sometimes this is desired (such as a surgical application, or when using a tool), but sometimes it is not (such as when turning a dial).
- Range of motion: **2**. While the Phantom's range of movements is limited, this is not a problem, as fine motor control tasks do not require more range of motion than the Phantom can provide.

#### **CyberGrasp:**

- Application of force and torque: **1**. The CyberGrasp can apply very few of the forces required for these tasks. It can only pull back on the fingers. This may be what is required for the particular action, but often will not be.
- Sufficient force and torque: **0**. Since the CyberGrasp cannot provide the majority of the forces and torques required, it cannot provide sufficient force and torque for those actions.
- Grasp simulation: **2**. The CyberGrasp can simulate the grasp of any solid object
- Range of motion: **2**. The CyberGrasp has the same range of motion as the user's hand.

All the force and torque types for this task are the same in terms of simulation effectiveness. They all receive an Excellent rating (9) for the Phantom, and a Moderate rating (5) for the CyberGrasp.

### **Significant arm strength tasks**

There are some problems with the simulation of tasks that require a lot of arm strength. The most obvious is that most, if not all, haptic devices cannot exert enough force or torque to simulate the amount of strength needed. There are serious safety issues with haptic devices that



can exert that amount of force (one bug in the program, and your arm could be broken). Thus, while one could learn the gross motor movements of how to do the task, the strength requirement cannot be simulated very well. There are ways to mitigate this, though – one way is to factor all forces down by a constant (or logarithmic) scale. Thus, light forces still feel light, and heavy forces still feel heavy. This would enable the user to realize that a task being performed would require more or less strength than the previous task being performed.

#### Phantom:

- Application of force and torque: **4**. The Phantom can provide all the necessary forces, if not in the required amounts.
- Sufficient force and torque: **0**. The Phantom cannot provide the force amounts required for a realistic simulation.
- Grasp simulation: **0**. The Phantom cannot provide the proper grasp simulation for these tasks. Any task that requires the user to hold a cylindrical shaft is a tool-assisted task, not a significant arm strength task.
- Range of motion: **1**. The other problem with these tasks is that they sometimes require a wide range of motion. For example, picking up a box from the ground requires a device that can move up to 3 feet. Not all tasks require a large range of motion, however – turning a stuck bolt with a wrench requires a much smaller range of motion.

#### CyberGrasp:

- Application of force and torque: **0**. The CyberGrasp cannot provide any additional forces or torques on the user, other than pulling the fingers back.
- Sufficient force and torque: **0**. Since the CyberGrasp cannot provide the appropriate forces and torques, it cannot provide sufficient force and torque for those actions.
- Grasp simulation: **2**. The CyberGrasp can simulate the grasp of any solid object
- Range of motion: **2**. The CyberGrasp has the same range of motion as the user's hand.

The total for the Phantom is **5**, yielding a moderate rating. This is slightly better than the CyberGrasp, which has a poor rating of **4**.

#### **Tactile (finger pressure) friction tasks**

These tasks have a lot in common with the fine motor control tasks. The difference is that these tasks require more haptic feedback than the fine motor control tasks.

#### Phantom:

- Application of force and torque: **4**. The Phantom's precision (discussed in the fine motor control category) allows these tasks to have an excellent realism when simulated in a virtual environment. The Phantom can apply all the forces and torques needed.
- Sufficient force and torque: **2**. As the force and torques needed are not large, the Phantom can easily supply sufficient force and torque.
- Grasp simulation: **0**. The problems occur when trying to address the grasp simulation. The Phantom can only simulate a cylindrical shaft, and cannot simulate the feel of something against different parts of the finger.
- Range of motion: **2**. The range of motion required for these tasks is not large, so the Phantom can easily provide that range of motion.

#### CyberGrasp:

- Application of force and torque: **1**. The CyberGrasp can apply very few of the forces required for these tasks. It can only pull back on the fingers. This may be what is required for the particular action, but often will not be.
- Sufficient force and torque: **0**. Since the CyberGrasp cannot provide the majority of the forces and torques required, it cannot provide sufficient force and torque for those actions.
- Grasp simulation: **2**. The CyberGrasp can simulate the grasp of any solid object
- Range of motion: **2**. The CyberGrasp has the same range of motion as the user's hand.

The Phantom comes in ahead in this category, receiving a good rating of 8.

#### Cooperative two-handed tasks

These tasks have some additional concerns that are not present in the other task categories. They must use two haptic devices, with double the computing capability. This exposes issues of synchronization, latency, etc.

As described earlier, each haptic device has its own strengths and weaknesses. The Phantom can provide better haptics forces, while the CyberGrasp has much better range, as it is not attached to a heavy base unit. In order to rate actions in this category, one must pick a pair of haptic devices to rate. Not surprisingly, we are considering a combination of the Phantom and a CyberGrasp.

#### Phantom and CyberGrasp:

- Application of force and torque: **3**. The CyberGrasp can simulate holding of an object very realistically, and this may be all that is required in the simulation (the CyberGrasp hand holds the object, and the Phantom does the action). However, the CyberGrasp

cannot simulate any other types of forces or torques. The Phantom can provide all the required forces and torques.

- Sufficient force and torque: **1**. The CyberGrasp cannot provide most of the required forces, unless the hand is holding an object. Some tasks (but not all) will require a significant amount of strength (which is why two hands are needed), which will be more than the Phantom can provide.
- Grasp simulation: **1**. The CyberGrasp can simulate the grasp of any solid object, but the Phantom is limited to simulating the grasping of a solid cylinder.
- Range of motion: **1**. The CyberGrasp has the same range of motion as the user's hand, but the Phantom's range of motion is limited.

### **Braced two-handed tasks**

Any task that requires the user to brace one hand is a task that requires a significant amount of strength, as otherwise the brace would not be needed. Bracing in a virtual environment is not easy without some physical armature present. The actual act of bracing in the virtual environment would probably have to be "faked", meaning that the computer would assume that if the hand were near the brace, then the brace is occurring.

#### **Phantom:**

- Application of force and torque: **4**. The Phantom can provide all the necessary forces, if not in the required amounts.
- Sufficient force and torque: **0**. The Phantom cannot provide the force amounts required for a realistic simulation.
- Grasp simulation: **0**. The Phantom cannot provide the proper grasp simulation for these tasks. Any task that requires the user to hold a cylindrical shaft is a tool-assisted task, not a braced two-handed task.
- Range of motion: **0**. The tasks that require a brace are going to require more range of motion than the Phantom can provide.

#### **CyberGrasp:**

- Application of force and torque: **0**. The CyberGrasp cannot provide any additional forces or torques on the user, other than pulling the fingers back.
- Sufficient force and torque: **0**. Since the CyberGrasp cannot provide the appropriate forces and torques, it cannot provide sufficient force and torque for those actions.
- Grasp simulation: **2**. The CyberGrasp can simulate the grasp of any solid object
- Range of motion: **2**. The CyberGrasp has the same range of motion as the user's hand.

Both the Phantom and the CyberGrasp receive a poor rating (4). As the Phantom is used more often, it is shown in the table that follows.

### **Manipulating deformable object tasks**

A consideration for these tasks is that the computer models that simulate the deformations are much more complex than they would be for rigid bodies, requiring significantly more computation time.

Recall that having the Phantom prod a deformable object is considered a tool-assisted task. This category is when you are interacting with the deformable object directly.

#### **Phantom:**

- Application of force and torque: **4**. The Phantom's precision allows it to simulate the forces and torques required for this category of tasks very well.
- Sufficient force and torque: **1**. Many tasks will not require significant strength; therefore, the Phantom can provide the appropriate amount of force and torque. However, some tasks will require more strength than the Phantom can provide.
- Grasp simulation: **0**. You are holding the (very solid) shaft of the Phantom, and not a deformable object. Thus, while the forces and torques can be realistic, the application of them (through the hard shaft of the Phantom) will not be.
- Range of motion: **1**. One can easily think of example simulations that both do and do not exceed the range of motion that the Phantom provides.

#### **CyberGrasp:**

- Application of force and torque: **0**. The CyberGrasp cannot provide any additional forces or torques on the user, other than pulling the fingers back.
- Sufficient force and torque: **0**. Since the CyberGrasp cannot provide the appropriate forces and torques, it cannot provide sufficient force and torque for those actions.
- Grasp simulation: **2**. The CyberGrasp can simulate the grasp of a deformable object.
- Range of motion: **2**. The CyberGrasp has the same range of motion as the user's hand.

The Phantom is rated higher with a moderate rating (6).

### **Tool-assisted tasks**

The simulation realism possible for tool-assisted tasks is quite good. Many believe that this category is what these haptic devices are best suited for.

#### Phantom:

- Application of force and torque: **4**. The Phantom can exert all six degrees of freedom (three force and three torque), so all forces and torques can be simulated.
- Sufficient force and torque: **1.5**. The amount of force that the Phantom can provide may not be sufficient for tool tasks, as a task that requires a tool is often something that requires a fair amount of force. However, these forces can often be scaled down while keeping the majority of the simulation realism.
- Grasp simulation: **2**. The shaft of the Phantom can simulate the handle of most tools (hammers, screwdrivers, wrenches, etc.).
- Range of motion: **1.5**. Although the shaft of the Phantom can move in any direction, the base unit is not particularly mobile. Most tools can be used within the Phantom's range of motion, however.

#### CyberGrasp:

- Application of force and torque: **0**. The CyberGrasp cannot provide any additional forces or torques on the user, other than pulling the fingers back.
- Sufficient force and torque: **0**. Since the CyberGrasp cannot provide the appropriate forces and torques, it cannot provide sufficient force and torque for those actions.
- Grasp simulation: **2**. The CyberGrasp can simulate the feel of a handle of a tool as well as the Phantom can.
- Range of motion: **2**. The CyberGrasp has the same range of motion as the user's hand.

The CyberGrasp yields a rating of **4**. While this is much less than the Phantom rating (**9**), the range of motion that the CyberGrasp allows may make this a better choice.

#### Multi-finger tasks

This category of tasks cannot be simulated realistically with all haptic devices. The Phantom, for example, requires you to hold a shaft – thus, there are no multi-finger aspects to it. The CyberGlove, however, is specifically designed for these types of tasks.

#### Phantom:

- Application of force and torque: **0**. Although the Phantom can apply any force or torque desired, it cannot apply different forces and torques to different features, which is what is required for this task category.
- Sufficient force and torque: **0**. Since the Phantom cannot provide the appropriate forces and torques, it cannot provide sufficient force and torque for those actions.

- Grasp simulation: **0**. The Phantom requires you to hold a cylindrical shaft, which does not apply different forces on different fingers.
- Range of motion: **1**. The ranges required for these types of actions are often not large, and thus can be simulated by the Phantom.

#### CyberGrasp:

- Application of force and torque: **4 or 0**. The CyberGrasp is capable of pulling the fingers backwards, which is the type of force in the Force I category. Note that it cannot push the fingers forward, but this type of force is rarely needed. Thus, an action like playing a musical instrument or typing on a keyboard can be simulated with great realism, as the CyberGrasp can pull back on the fingers when needed. However, the CyberGrasp cannot provide the type of force required for the second force category, for this would be an action such as dragging a finger along a desk (or other surface with friction). The force applied is not in line with the finger; it is pulling to the side. Also, the CyberGrasp cannot provide any sort of torque. Thus, for the first criteria (application of force and torque), this task category receives a 4 for the Force I category, and a 0 for the other force/torque categories.
- Sufficient force and torque: **2 or 0**. The CyberGrasp can provide sufficient force for the Force I category (a rating of 4). Since it cannot provide the correct force and torque for the others, it obviously can't provide sufficient force or torque (a rating of 0 for the other 4 force/torque categories).
- Grasp simulation: **2**. The CyberGrasp can simulate the grasp of any solid object
- Range of motion: **2**. The CyberGrasp has the same range of motion as the user's hand.

The CyberGrasp clearly rated better for these tasks, receiving an excellent rating (**10**) for the Force I category, and a poor rating (**4**) for the other four categories.

Note that some multi-finger tasks, such as turning a dial, could be simulated by the Phantom by having the task become a tool-assisted task (the shaft of the Phantom is connected to the dial).

### 3. Simulation Effectiveness Summary

Table 3 summarizes the effectiveness of simulating the actions with today's haptics technology. Each table entry has three parts. The first is the rating (excellent, good, poor, etc.). The second is the numerical score that task received based on the above criteria. The last part (on the second line of each table entry) is the breakdown of the ratings the task received for each of the separate criteria. The are, in order from left to right, application of force and torque, sufficient force and torque, grasp simulation, and range of motion.

	Force Only I	Force Only II	Torque Only I	Torque Only II	Force & Torque
Fine motor control	Excellent (9) (4, 2, 1, 2)	Excellent (9) (4, 2, 1, 2)	Excellent (9) (4, 2, 1, 2)	Excellent (9) (4, 2, 1, 2)	Excellent (9) (4, 2, 1, 2)
Significant arm strength	Moderate (5) (4, 0, 0, 1)	Moderate (5) (4, 0, 0, 1)	Moderate (5) (4, 0, 0, 1)	Moderate (5) (4, 0, 0, 1)	Moderate (5) (4, 0, 0, 1)
Tactile (finger pressure) friction	Good (8) (4, 2, 0, 2)	Good (8) (4, 2, 0, 2)	Good (8) (4, 2, 0, 2)	Good (8) (4, 2, 0, 2)	Good (8) (4, 2, 0, 2)
Cooperative two-handed tasks	Moderate (6) (3, 1, 1, 1)	Moderate (6) (3, 1, 1, 1)	Moderate (6) (3, 1, 1, 1)	Moderate (6) (3, 1, 1, 1)	Moderate (6) (3, 1, 1, 1)
Braced two-handed tasks	Poor (4) (4, 0, 0, 0)	Poor (4) (4, 0, 0, 0)	Poor (4) (4, 0, 0, 0)	Poor (4) (4, 0, 0, 0)	Poor (4) (4, 0, 0, 0)
Manipulating a deformable object	Moderate (6) (4, 1, 0, 1)	Moderate (6) (4, 1, 0, 1)	Moderate (6) (4, 1, 0, 1)	Moderate (6) (4, 1, 0, 1)	Moderate (6) (4, 1, 0, 1)
Tool-assisted tasks	Excellent (9) (4, 1.5, 2, 1.5)	Excellent (9) (4, 1.5, 2, 1.5)	Excellent (9) (4, 1.5, 2, 1.5)	Excellent (9) (4, 1.5, 2, 1.5)	Excellent (9) (4, 1.5, 2, 1.5)
Multi-finger tasks	Excellent (10) (4, 2, 2, 2)	Poor (4) (0, 0, 2, 2)	Poor (4) (0, 0, 2, 2)	Poor (4) (0, 0, 2, 2)	Poor (4) (0, 0, 2, 2)

**Table 3: Taxonomy Task Simulation Effectiveness Summary**

Note that some example actions appear multiple times in Table 2, and their corresponding entries in Table 3 do not always match. The bayonet connector, for example, is an excellent way to simulate a fine motor control or tactile friction task (for force and torque), but a poor way to simulate a multi-finger task (for force and torque).

The table reflects the fact that haptic devices were created with specific tasks in mind. Fine motor control tasks, for example, were one some of the tasks considered in the development of the Phantom. Multi-finger tasks (Force I only) were the type of tasks considered in the creation of the CyberGrasp. Some tasks do not have haptic devices that can properly simulate them – for example, the braced two-handed tasks and most of the multi-finger tasks. As haptics technology continues to improve, we expect to see better effectiveness in all the areas, including the areas that received a Poor rating.

## B. Testing Requirements

The specific testing requirements are described before the table – each action section describes which forces and torques are needed to simulate that particular action. An overview of



how useful each of the three technologies (computation and visualization only, contemporary haptic technologies, and maximal devices) is described below.

Note that since this entire report's focus is on using haptics for aircraft maintenance validation, those aspects are not gone into detail in this section. Only the non-haptic version is examined in depth here.

## 1. Computation And Visualization Only

The goal of this study is virtual task validation. There are two general approaches to this goal:

1. Interactive user task attempts and analysis
  - a) Using visual and haptic feedback
  - b) Using visual feedback only
2. Non-interactive task attempts and analysis

Although the basis of this study is case 1a, this division emphasizes that alternatives to haptics are possible and must be explored even if only as experimental controls. Thus we will address 1b and 2 in this section.

Any claims about the veracity or usefulness of haptics simulations for maintenance actions ought to be measured against similar tasks executed without haptic feedback. A baseline non-haptic simulation (case 1b) would involve a visual interface but use manual interaction devices without haptic feedback. In general, such devices would be standard computer input devices such as a mouse, keyboard, joystick, or trackball. A CyberGlove, although not very common for individuals (but somewhat common in labs) can also be included. Manipulating these devices would cause CAD objects to move interactively (real-time) on screen. Purely visual feedback on task progress would be displayed. Such an arrangement would typically include camera controls so that the user view could be readily changed to any suitable position and visual feedback for at least collision detection. In addition, visual feedback might be available to monitor forces and torques needed relative to human maintainer capabilities. In an ideal situation, such limitations would actually be used as constraints on the allowable movements executed by the user. At present, we know of no software tools that enforce such constraints during interactive manipulation. This is a possible area for future algorithm research and development.

In case 2, "pure" computation would be used to establish task validity. What this means is that a computer simulation would need to establish that a given maintenance task could or could not be performed. There are four possible outcome cases:

- A. The task is physically possible and humanly possible by any "typical" aircraft maintainer.



- B. The task is physically possible but unreasonable to expect from a "typical" aircraft maintainer (e.g., insufficient strength).
- C. The task is physically impossible due to human limitations (of any maintainer).
- D. The task is physically impossible due to physical limitations (part is just inaccessible or not extractable).

Clearly this problem is complicated by the need to characterize aircraft maintainers and their statistical capabilities with respect to anthropometry, dexterity, strength, and skill. (Our haptic study does not overtly address some of these variables though they are critically important to assess situations A, B, and C.) Existing human form models do cover some of these critical variables. . Independent of how well or not such human form models parameterize this space, it is crucial to note that any interactive simulation based on a real user will represent solutions based on a sample set of one, and thus will not provide any more broad parameterization than existing human form models: a single user is not a statistically useful datapoint for task validation (with or without haptics!) except possibly for cases C and D. Human factors experimenters will typically use several subjects to assess task validity for cases A and B. Such multiple subject tests are clearly possible and desirable for statistical purposes.

Increasing the number of subjects that attempt to validate a task increases both cost and set-up time. Human form models address this accommodation problem directly by allowing the user to manipulate or test task validity with multiple models computed or selected from known anthropometric populations (such as aircraft maintainers). It is not known at this time whether virtual interactive simulations with haptic feedback will be more or less costly than non-haptic simulations or tests on actual physical devices (mock-ups or the actual aircraft).

Returning to the outcomes A-D above, task validation requires establishing which one obtains given a specific task. Can "pure" computation (case 2) play a role? We believe the answer is affirmative, but it depends on developing new software approaches to human modeling. The major issues are:

- Reach algorithms must find access paths in confined spaces or determine that no solution can be found.
- These algorithms must take into account body size, articulation, joint limits, soft tissue deformations, tool handling, and clothing restraints and thickness in order to make accurate reachability assessments.
- These algorithms must respect human torque and strength limitations.
- These algorithms must also address multi-point bracing and contacts as leverage to strategize and complete tasks.

Although no existing reach algorithm meets all these goals, it remains a desirable research objective. We believe that such an algorithm is possible but it will take dedicated development. Funding opportunities should consider this option in parallel to interactive and

haptic feedback methods, since these other operational approaches (including haptics) cannot presently satisfy the overall validation goal either.

## 2. Contemporary Haptic Technologies

Many of the actions in the table can be successfully simulated with existing haptic technologies. Specifically, all the "excellent" and "good" entries in Table 3 are appropriate for simulation with current haptic devices. The "moderate" entries would not be useful as a solitary haptic action simulation, but might prove useful as one of many haptic actions performed to complete a complex task.

## 3. Maximal Devices

A maximal device, a full exoskeleton, could conceivably simulate all of the actions in the table. An exoskeleton would be able to provide significant strength resistance to a user's actions, which is the majority of the "moderate" entries in the table. An exoskeleton designed for haptic simulation purposes (as opposed to designed for strength enhancement, such as GE's exoskeleton, described in section 3.5.1, Virtual Reality Research Review, under the Current Haptics Technology sub-section) would most likely have the capability to provide forces and torques to the individual fingers. Such a system would provide the best possible haptic simulations possible today. Its financial cost and the difficulties developing such a system, however, make it impractical to obtain or use one.

## C. Technical Hurdles

Although each action in the taxonomy was assessed for technical hurdles, the result was a series of difficulties that can happen to any and all the actions. Thus, they are presented together.

One of the main hurdles of haptics research is the computer execution speed. The Phantom requires functions that execute under a specific time limit (it calls these functions 1,000 times a second). This causes problems with complicated models, which cannot determine if the Phantom arm has intersected the object within the required amount of time. This prevents the use of complicated models for haptics interaction with the Phantom. This problem is mitigated by the increasing speed of processors, but not enough. One solution would be for the Phantom to allow for slower executing functions with the trade-off of less realism, and allow the developer to determine what the appropriate balance is. Processor speed also limits graphical realism, especially when a significant amount of the processor's computing power is determining the haptics aspect of the simulation.

Buggy libraries! The libraries provided by the manufacturers are often riddled with bugs. This is partly due to the fact that the device manufacturers are continually updating the libraries with new features, and partly because the field of haptics is still in its infancy. Neither of these is much of a consolation to the developer running into those bugs, of course.

Many of the libraries do not provide collision detection routines, which are obviously essential for haptics (as the haptic device has to collide with something in the virtual world). This requires the developer to have to write a lot of collision detection code, and spend time debugging it, when it could be included in the libraries. Of course, then the developer couldn't debug it if they got the pre-compiled library with buggy collision detection code.

A closely related aspect to the buggy libraries is the poor documentation that for the libraries that exist. This only makes getting the libraries to work that much harder.

The limited range of haptic devices, and their (relatively) limited function (compared with the function of the real world) could be considered a device limitation or a technical hurdle. As the field matures, and more companies start to develop haptic devices, this problem will lessen. The CyberGrasp, for example, cannot apply torque to the user's hand or fingers.

#### D. Benchmarks and Comparisons Among Haptics Platforms

Comparison of simulations using different haptic devices is a difficult task to perform. If the haptic devices are similar, such as two robot arms, then the comparison is made much easier. But that is not the case with a comparison between simulations with a Phantom and a CyberGrasp.

Comparisons between a haptic environment and the real world simulation are also difficult to quantify. Obviously the simulation will not be an exact replica of the real-world simulation. The problem is what criteria should be chosen for the comparison. With a computer graphics picture, there are image-processing techniques that, using various heuristics, can compare a computer model with its real-world counterpart.

Both of these types of comparisons have the same solution: ask the user. After a participant uses a haptic environment, they can answer certain questions such as the following. To allow for a numerical score, the questions would be answered on a scale of 1-10.

- How realistic did the graphics of the simulation seem? (1 = totally unrealistic, 10 = completely realistic)
- How realistic did the haptics of the simulation seem? (1 = totally unrealistic, 10 = completely realistic)
- How realistic did the simulation feel compared to the real-world situation? (Only valid if they have done the real world action) (1 = totally unrealistic, 10 = completely realistic)
- How comfortable would you feel performing the task in the real world? (1 = not comfortable at all, 10 = completely comfortable)
- How much did you feel you learned about this task through the simulation (1 = nothing, 10 = as much as one could learn about the task)
- How realistic did the grasp simulation seem? (1 = totally unrealistic, 10 = completely realistic)

- How realistic did the magnitude of the forces and torques seem? (1 = totally unrealistic, 10 = completely realistic)
- How helpful did you feel this simulation was? (1 = not helpful at all, 10 = very helpful)

There will most likely be other questions, specific to the particular task being simulated.

A proper study would require an appropriate number of human subjects to yield statistically valid results, and control measures, to determine if the change in behavior can be attributed to the haptic interaction. For example, a group of people who did not perform a disassembly task under the simulation would perform the task in real life, and their results would be compared to those who went through the simulation. This would provide hard data as to the effectiveness of the simulation.

Another factor to consider is sample population. While it will undoubtedly be easier to get your fellow lab mates to perform the tests, if the simulation is designed for novice computer users, the results would not be generalizable.

Note that these particular simulations might not be a single task. A series of haptic actions (such as changing a tire) would yield much more interesting results than just a single haptic task (Can you push this button? How realistic did pushing that button feel? How comfortable do you feel pushing buttons now that you pushed the button in this simulation?).

## E. Evaluation Results

The results for the evaluation are in section XI (Demonstration of the Assessment & Validation Approach) in the Analysis section (section D). To exercise the evaluation procedure, we evaluated the simulation on one person. A more rigorous evaluation is needed to accurately assess the impact of haptics on maintenance simulation. Also, our evaluation was for a single haptic action (a bayonet connector). Everybody has used a bayonet connector in one form or another (i.e. a medicine bottle). The real value of haptic simulations comes into play when simulating multiple actions in succession.

## VI. Failure Condition Analysis

*Goal: To summarize and analyze the failure conditions identified during the assessment to determine which ones lend themselves to automation in the environment.*

A number of the requirements for this section are found in other places in this report. The classes of tasks that will not work well in a haptic simulation are described in the previous section, and summarized in the taxonomy task evaluation effectiveness summary table. Following that table, the three haptic configurations (no devices; current affordable devices; and a full exoskeleton) are used to show which lend themselves to an optimal simulation for a given task, and which do not. The success and failure conditions are discussed in this section, as well as in the previous section under the Benchmarks and Comparisons Among Haptics Platforms sub-section.

### A. Error Types

The list of errors presented in this section is not an exhaustive list of all errors that can occur in a haptic environment. An important distinction to remember is that the errors listed below are for *individual* haptic actions. Thus, if a user performs a series of haptic actions in the wrong order (use a jack to raise a plane, remove a flat tire from a plane to fix it, then try to lower the jack without replacing the tire), the errors below will not catch these cases. Note that the user may have performed each of the haptic operations involved with removing the tire perfectly! The simulation environment must handle these sorts of errors, which are "beyond" the errors encountered with individual haptic actions. The reason is that these errors are dependent on the simulation environment, and not on the haptic actions. To be able to differentiate between the two, we are designating them "haptic action errors" (the errors listed below) and "simulation errors". Note also that some of the errors listed below can also be simulation errors. One example is the time limit - this error can be both a haptic action error (the user took too long to screw that one bolt into place) or a simulation error (the user took too long to screw those seven bolts into place).

We have developed a list of possible individual haptic action errors, which is enumerated below.

#### 1. User Errors:

- a. User exerts too much force or too little force
- b. User exerts too much torque or too little torque
- c. User exerts force in the wrong direction
- d. User exerts torque in the wrong direction or around a wrong axis
- e. User exerts force too early or too late

- f. User exerts torque too early or too late
  - g. User exerts force for too long a time period or too short a time period
  - h. User exerts torque for too long a time period or too short a time period
  - i. Right action on wrong object (all actions have this error)
  - j. Wrong action on right object (all actions have this error)
  - k. Actions done in wrong order
2. Device Limitations:
- a. Maximum force or torque that the haptic device could safely provide is insufficient to model the physical system.
  - b. Maximum movement allowed by the device is insufficient to model the required action.
  - c. The haptic device is not mobile, thus preventing proper reproduction of the physical system.
  - d. The grasp requires friction or tactile feedback not provided by the haptic device.
3. Insufficient access: The task required manipulating the hand/arm in such a way that environmental constraints (collisions) between the arm and other objects would have prevented the motion.
4. Insufficient function: The task would have caused a response (e.g., a hazard, vibration, shock, temperature change, fluid release, etc.) in the physical system that could not be represented via haptic feedback.
5. Ability Limitations: User could not complete the operations because:
- a. Object to operate on is not visible currently (some other operations might be needed to make it visible).
  - b. Object is not in a safe operable status (e.g., too hot, still running, etc.).
  - c. Object is too big or too heavy (manipulation might require tools or more people)

User errors are errors that could conceivably occur due to lack of training or accidents. Device limitations are dependent on the particular haptic device being used, and not on the action being performed. Insufficient access errors depend on the virtual reality environment, and not the haptic action (there may or may not be a wall blocking the way to perform a particular haptic action). Insufficient function errors are similar to device limitation errors, but there is a distinction. Insufficient function errors are responses from the computer environment that cannot be simulated by any haptic device (temperature change, fluid release). Device limitation errors are errors that cannot be simulated by that particular haptic device (not enough strength, not enough range of movement, etc.), but might be able to be simulated by other haptic devices.

Ability limitation errors are similar to user errors, but there is also a distinction between them. The difference is that the ability limitation errors are cases where the user could not perform the desired action, even if fully trained (it is beyond the user's strength to turn that particular bolt without a tool), whereas user errors are preventable with training. Ability limitations, like insufficient access errors, depend on the simulation environment, and not the haptic action performed.

## B. Error List

Each cell in the task taxonomy error list (Table 4) will only list the possible user errors that can occur, as the other errors can occur for any action, as described above. To save space in the error list table, the errors are referred to by a single letter (a-k).

Detecting user errors automatically depends on having a model of the correct actions against which user actions are compared. The PAR framework can serve this function, since the PAR for the task action would store the parameters, paths, forces, or torques relevant to the task. User actions deviating from nominal task parameters can be flagged and visually or audibly signaled to the user.

	Force Only I	Force Only II	Torque Only I	Torque Only II	Force & Torque
Fine motor control	<i>a, c, g, i, j</i>	<i>a, g, i, j</i>	<i>b, d, h, i, j</i>	<i>b, d, h, i, j</i>	<i>a-k</i>
Significant arm strength	<i>a, c, g, i, j</i>	<i>i, j</i>	<i>b, d, h, i, j</i>	<i>b, d, h, i, j</i>	<i>a-k</i>
Tactile (finger pressure) friction	<i>a, c, g, i, j</i>	<i>e, i, j</i>	<i>b, d, h, i, j</i>	<i>b, d, h, i, j</i>	<i>a-k</i>
Cooperative two-handed tasks	<i>a, c, g, i, j</i>	<i>c, i, j</i>	<i>b, d, h, i, j</i>	<i>b, d, f, h, i, j</i>	<i>a-k</i>
Braced two-handed tasks	<i>a, c, g, i, j</i>	<i>i, j</i>	<i>b, d, h, i, j</i>	<i>b, d, h, i, j</i>	<i>a-k</i>
Manipulating a deformable object	<i>a, c, g, i, j</i>	<i>c, i, j</i>	<i>b, h, i, j</i>	<i>d, h, i, j</i>	<i>a-k</i>
Tool-assisted tasks	<i>a, c, g, i, j</i>	<i>a, c, g, i, j</i>	<i>b, d, h, i, j</i>	<i>b, d, f, h, i, j</i>	<i>a-k</i>
Multi-finger tasks	<i>a, c, g, i, j</i>	<i>g, i, j</i>	<i>i, j</i>	<i>b, d, h, i, j</i>	<i>a-k</i>

Table 4: Task Taxonomy Error List



## VII. Validation Methodology Development

*Goal: To develop a methodology for performing validation for maintenance procedures using an SMG virtual configuration, including descriptions for set-up, execution and analysis.*

During our research in this area, it was realized that a decision tree would not be the most effective way of creating a methodology for performing procedure validations. Each node in the tree could have countless branches. One example is the haptic device category being used – there are dozens of them out there, and they are likely to change as technology advances. Thus, there are simply too many branches to provide a feasible tree. Instead, we have presented various considerations in a checklist like format, which will still be valid as technology progresses.

### A. Simulation Development and Setup

#### Primary Concerns

A combination of factors described below will determine if a particular task is appropriate for haptic simulation. However, there are certain concerns that will dictate all the aspects of the haptics simulation development and set. These concerns are discussed more below. These primary concerns include:

- Haptic task being performed – how suitable is it for haptic simulation?
- Haptic devices being used – is the device suitable for the given task? Or does a new device have to be purchased?
- Development time and costs

#### General Concerns

- Purpose: The purpose and goals of the haptic simulation needs to be fully understood. For the purpose of validating technical manuals, the simulation must have available a large suite of haptic actions to cover the range of tasks and maintainer skills. All validation simulations require a reasonable level of realism in space (geometry, physical layout, and assembly structure), movement (mechanical properties, degrees of freedom, and fastener constraints), and maintainer (human capabilities, strength, and access).
- Existing forces and torques: One needs to determine the particular forces and torques that exist in the environment, and how the user (via his haptic equipment) will interact with them. This includes the forces and torques that the haptic interaction will include. It may be the case that there is only one object in the simulation that one can interact with, but



most likely there will be many. One should consider whether existing (fixed) objects in the environment could be used to react forces exerted by the maintainer. For example, in space operations stable foot restraints are essential for proper task execution. Even on the ground, various aircraft elements may be used to provide a base or support for manual actions. Friction is still another consideration – it is an important aspect in simulating object interactions and is essential for maintaining realism.

- Haptic action being performed: The particular taxon for the task type should then be determined. This will yield a lot of useful information; most importantly being how feasible the action is to being simulated in a haptic environment (from the taxonomy task evaluation effectiveness summary tables). Other information can be obtained from knowing the category, such as the possible user errors (from the taxonomy table).
- Haptic device being used: The particular device being used brings with it its own set of benefits and drawbacks. The Phantom, for example, allows for a lot of precision, but the user is always grasping the shaft of the robotic arm. The CyberGrasp allows for individual finger force application, but does not allow for torque to be applied in any manner. No haptic device will be perfect for all simulated tasks. Often, the choice of the haptic device being used will be dictated by those available, or by the haptic action being performed if more than one device are available. These devices can be quite expensive (\$25,000 is the cost of the CyberGlove with the CyberGrasp), so financial considerations may also play into this decision.
- Host platform: The two main choices here will be a Microsoft Windows platform or a UNIX-based platform. This choice may be determined by the libraries that accompany the various haptic and VR equipment that the developer has on hand (many companies only release drivers for a limited range of platforms).
- Geometric model: The model of the haptic objects within the environment will have to be decided on. While CAD datasets are common, these are often the cause of performance decrements due to their complexity. Likewise, deformable models may cause performance problems. Many haptic devices (such as the Phantom) require the various functions to have a *very* fast execution time. (The Phantom calls these functions 1,000 times a second, and the computer must also compute the 100 graphical screen updates with the left-over processor time in-between those function calls.) Thus, using a complex CAD dataset would not be feasible. Another consideration is what functions the haptic device needs to be able to call. Some devices require line segment intersection routines (“at what points does this line intersect your object?”), which can be difficult to write for raw CAD datasets.
- Physical attribute model: Even if the CAD models for the assembly components are available, several additional considerations may be needed to make these models amenable to haptic simulations:
  - Geometry files: CAD geometry files may need to be converted to the chosen simulation format. Surfaced (boundary representation) models are the most convenient for display and haptic simulation, yet many CAD systems use

Constructive Solid Geometry (CSG). Converting CSG data to surfaces is not difficult, but it does increase the size of the dataset. Conversion from smooth curved surfaces into polygon models may also result in approximations that yield incorrect physical simulations: e.g., a cylindrical rod is free to rotate and slide inside a hole, but if approximated by a polyhedral rod rotation may be disallowed.

- Assembly and structure data: This will be needed so that manipulable components may be identified and segmented. Thus articulated joints and fasteners must be identified and their geometric degrees of freedom noted,
- Physical simulations: Since the target is haptics, physical simulations are needed. Computing forces and torques requires that the objects include physical attributes such as mass, moments of inertia, spring constants, friction coefficients, and force/torque constraints. Consider, e.g., that a bolt in an assembly must be identified as a separate movable object, must be tagged as having a left- or right-handed thread, and perhaps having a preferred insertion torque. Conceivably the geometry structure might provide the separability, the product database may provide the handedness, and some maintenance procedure might cite the torque requirement. For deformable models, local constraints on flexion, folding, bending, shearing, and tearing might be needed but are very likely available (if at all) only in specifications or documentation based on the original material.
- Graspable sites: Graspable sites for maintenance activities are rarely noted in geometric models. The maintainer may need to understand what parts are handles, hooks, grasping holes, or release pins or levers. Rigid objects suitable for reacting forces would also be needed as we noted above. Maintainer aids such as tape (to hold cables out of the way) or other unrelated parts used only to restrain flexible components will be difficult to identify from the CAD geometry or the instructions themselves: they may only be gleaned from observation (training) or actual experience.
- Graphical environment: The aesthetics of the virtual world need to be taken into account. Many haptic actions will not need a visually compelling environment, but it should be detailed enough that the user can suspend their disbelief during the simulation. While one can create a haptic simulation that does not have any visual aspects whatsoever (i.e., a blank screen), that is usually not the case with most haptics development. If the graphics generation can be rendered on a different processor or computer than the haptics are being performed on, this will help with performance issues (as long as the presentation of both modality is accurately synchronized). If realism is desired, a head mounted display should be used. A note here is that the (hopefully relatively simple) geometric model that the haptic device uses does not have to be the same as what the user sees. A simulation can have a lot of graphical detail while keeping the underlying geometry haptic model simple. This allows for quicker haptic execution (as the model is simpler) while still allowing for graphical realism.
- Development environment: The development environment will play a major factor in how long it takes to create the simulation, execution time, etc. Languages like Java have

a lot of code libraries, but are slow in execution. Languages like C and C++ have very rapid execution times, but are more prone to programming bugs, and often do not have as good a library of useful code. As the most important factor is usually execution time, C and C++ seem to be the overwhelming choice. However, other languages (such as Java) are rapidly approaching C and C++ in execution time benchmarks. The code libraries are usually written in C and C++, which is another factor to consider. Many languages can call C and C++ library functions, however. The particular compiler will determine a number of aspects, including which debugging tools are readily available, and how portable the code is.

- Development personnel: The development of haptic simulations is not trivial. Consideration should be placed into how many people will be devoted to developing the code. There are often steep learning curves associated with learning a new haptic device. How much time is going to be required from each developer is dependent on a number of factors, including developer experience, the libraries available, the source code available, etc. Following good software engineering practices, methods to split the tasks among the various developers should be explored. This will also affect financial concerns, as developers are rarely free.

## B. Simulation Execution

The simulation is created by writing code that applies the internal and external forces and torques to a geometric model. Internal forces arise from springs, friction, (possibly) deformations, and reaction forces that prevent collisions and disallowed penetrations; external forces come from gravity and user actions. The physical simulation must interpret these in real time, as the haptic interfaces must present valid resistive forces and torques to the user at a high frequency – 1000 times a second or more. The rate-determining step in a complex physical simulation is the collision detection and response. As the complexity of the geometry increases the cost of collision detection increases even faster. Methods exist to reduce the computational costs of collision detection, usually by pre-processing the geometry into various bounding boxes and levels of detail to reduce the number of collision checks. Neither resource is likely to be provided in the existing CAD model and may therefore need to be added into the simulation environment after the CAD data and attributes are input.

The physical simulation itself entails using the input forces and torques to compute the accelerations of every object. Since object mass and moments of inertia are known (or added into the CAD data), linear and angular accelerations may be computed by applying  $F=ma$  ( $a=F/m$ ) formulas. From these accelerations, velocities and positions are computed by numerical integration over the frame rate (time). Some codes exist for this step (such as WorkingModel, Washington State University, etc.) but none seem to allow for the major geometric complexities of manipulating complex parts in confined spaces. Further investigation is needed to establish computing requirements and complexity bounds on physical simulation capabilities in the maintenance validation environment

### C. Evaluation Design

After the simulation and the associated haptics environment have been developed, a human subject runs through the simulation. The primary considerations for evaluation design are what subjects should run through the simulation and what sorts of data should be collected for subsequent analysis. We will examine the latter in the next section. Here we consider the subject choices.

To get a wide variety of human body shapes, many human factors researchers use a 5<sup>th</sup> percentile Japanese woman for the small end of human figure sizes, and a 95<sup>th</sup> percentile American male as the large end. To truly see if the haptic action is performable by a wide range of individuals, similar extremes should be tested. Unfortunately, strength is not necessarily correlated with body size. It is therefore awkward to limit haptic analyses to a small number (e.g., 2) of individuals. It is difficult to identify what a "weak" person is able to do and what a "strong" person is capable of. (Some human form simulation packages use strength data derived from empirical data, such as the aircraft maintainer strength model used in *CrewChief* or the NIOSH lifting data used in *Jack*. In general, these systems are not robust for complex or novel task actions.) For empirical haptic validations each evaluation should attempt to sample data from at least two individuals of roughly comparable anthropometry but of differing strength, hence at least four people. Another factor to consider is what type of training a subject should have prior to running through the simulation. For example, is the simulation for trained aircraft repair technicians, or for novices who are learning how to repair an aircraft for the first time?

If the evaluation requires physical components to simulate rigid aircraft parts that may be used to react body, leg, or opposite hand forces, then these components must be constructed as part of the evaluation set-up. Adjustable bars, beams, panels, or barriers may need to be assembled to create a physically constraining space in which the manual haptic actions can be performed. Without these constraints the user may assume a body position or pose which is unattainable in the physical context of the aircraft. Since validation is the target, as much physical realism as possible ought to be provided. The creation of such a flexible real constraining space around the virtual haptic space may be quite challenging to arrange. Later we will propose a possible surrogate for at least some of this physical context.

The evaluation of technical documents requires that they be accessible and modifiable based on the experiments. Electronic manuals, such as IETMS, have the potential to serve as interactive databases for instructions. Text instructions may be converted into action representations (e.g., PAR) and then used to control kinematic or, eventually, haptic (dynamic) simulations. Since the electronic manuals are interactive, annotations can be added to difficult instructions, or modifications made to instruction sequences, choices, orderings, or details as needed [BBE+00]. The primary issue is how the changes are made. With existing technology it is most feasible to simulate the instructions, watch for simulation or user errors, and manually annotate or modify the instructions. In general, no simple method will be able to invent novel approaches to ameliorate hard maintainability problems, but current techniques can likely flag hard to understand or ambiguous instructions, note omitted steps, show unachievable goals, generate cautions or warnings, or suggest alternative camera views for illustrations.

## D. Analysis

There are two types of analysis that can occur: subjective and objective. The reason for this is that there are two ways to judge a haptics simulation. The first is by the realism or believability (the subjective part). The second is whether the task (as described in the technical manual) is possible as is, or whether quantifiable changes are needed in either the task description or the design itself: this is the essence of *maintainability*.

### Subjective Analysis

The best method to analyze and validate the realism of a haptic simulation is through human subject testing. This was discussed previously in the Assessment of Virtual Maintenance Actions section (under the Benchmarks and Comparisons Among Haptics Platforms subsection). A number of users should test the simulation, and rate it based on the various questions described before (which are also reproduced below). Depending on how much time and effort is spent on the human subject testing, the reliability of the analysis can be very high, especially if the test subjects can be compared to non-test subject in a real-world example of the simulation.

For the analysis, we used a Likert Scale to generate a numerical score. A Likert scale most often uses a 1-5 rating, where each number corresponds to the following.

1. Strongly unfavorable to the concept
2. Somewhat unfavorable to the concept
3. Undecided
4. Somewhat favorable to the concept
5. Strongly favorable to the concept

After a participant uses a haptic environment, they can answer certain questions such as the following. There will most likely be other questions, specific to the particular task being simulated.

- How realistic did the graphics of the simulation seem? (1 = totally unrealistic, 5 = completely realistic)
- How realistic did the haptics of the simulation seem? (1 = totally unrealistic, 5 = completely realistic)
- How realistic did the simulation feel compared to the real-world situation? (Only valid if they have done the real world action) (1 = totally unrealistic, 5 = completely realistic)
- How comfortable would you feel performing the task in the real world? (1 = not comfortable at all, 5 = completely comfortable)

- How much did you feel you learned about this task through the simulation (1 = nothing, 5 = as much as one could learn about the task)
- How realistic did the grasp simulation seem? (1 = totally unrealistic, 5 = completely realistic)
- How realistic did the magnitude of the forces and torques seem? (1 = totally unrealistic, 5 = completely realistic)
- How helpful did you feel this simulation was? (1 = not helpful at all, 5 = very helpful)

### **Objective Analysis**

The objective analysis will include a number of factors, some of which can be determined by the simulation itself, and others that can be determined objectively by a human operator or the simulation subject. These factors are the error list, described in the previous section. Whereas the previous section was simply listing which errors were encountered, the analysis section will analyze why those errors occurred, and potentially how to fix them. By going through the following assessment, the developer will gain insight as to how to better design the task description or the aircraft design.

As we noted, it may be possible to automate the detection of certain user errors by converting the task descriptions to PARs and comparing user actions against PAR nominal parameters. This would be an ambitious researchable topic as there is yet no simple connection between haptic simulation and a database such as a Product Data Management system that would store assembly structure, device use, and fastener characteristics. Even more challenging would be determining the appropriate remediation for such errors. While some alternatives can be easily proposed and tested, such as changing in the maintainer's access path or showing obstructions that should be moved out of the way in advance, more general engineering design solutions such as developing an assistive tool, reconfiguring the access space, redesigning the assembly, or inventing new structures are clearly out of the scope of the present investigation.

1. Which user errors did the subject encounter? As these are often errors of the human subject, they can usually be corrected by education (i.e. by teaching the user how to properly perform the action) rather than redesigning the simulation.
  - a) User exerts too much force or too little force
  - b) User exerts too much torque or too little torque
  - c) User exerts force in the wrong direction
  - d) User exerts torque in the wrong direction or around a wrong axis
  - e) User exerts force too early or too late
  - f) User exerts torque too early or too late
  - g) User exerts force for too long a time period or too short a time period



- h) User exerts torque for too long a time period or too short a time period
- i) Right action on wrong object (all actions have this error)
- j) Wrong action on right object (all actions have this error)
- k) Actions done in wrong order

## 2. Device Limitations:

- a) Was the maximum force or torque that the haptic device could safely provide insufficient to model the physical system? If so, can all the forces that are required for the simulation be scaled down while still providing a realistic simulation? While not as realistic as it would be otherwise (with the forces of actions not scaled down), it may still allow the users to get the general idea. This would allow the user to get a feel for the actions, without requiring all the strength that the real world action requires. If not, then either a new haptic device needs to be acquired, or the developers should look into simulating different haptic actions.
- b) Was the maximum movement allowed by the device is insufficient to model the required action? All haptic devices have their own set of limitations, and the goal is to determine which actions a particular device best simulates. One may have to change the haptic device being used, or the action being simulated, if the device proves insufficient for the current task simulation.
- c) Was the haptic device not mobile, thus preventing proper reproduction of the physical system? If so, can it be made more mobile? Perhaps by putting it on a platform that has more range of motion (by casters, a robotic arm, etc.). If not, then perhaps the ranges of haptic actions in the simulation can all be scaled down to allow for a simulation within the bounds of the haptic devices. While not as realistic as it would be otherwise (with the ranges of actions not scaled down), it may still allow the users to get the general idea.
- d) Does the grasp require friction or tactile feedback not provided by the haptic device? Can the haptic device provide this feedback with further simulation development? Is it absolutely necessary to provide this friction?

- 3. Insufficient access: The task required manipulating the hand/arm in such a way that environmental constraints (collisions) between the arm and other objects would have prevented the motion. The objective in this case is to determine what would occur in a real world situation. The haptics simulation may very well have performed exactly as intended.

4. **Insufficient function:** Would the task have caused a response (e.g., a hazard, vibration, shock, temperature change, fluid release, etc.) in the physical system that could not be represented via haptic feedback? If so, is it possible to provide that type of response through the other two sensory channels (visual and auditory)? Users can often suspend their disbelief for certain aspects of the simulation, this allowing a graphical and auditory representation of such a hazard to simulate the occurrence of the hazard itself.
5. **Ability Limitations:** User could not complete the operations because:
  - a) Was the object to operate on not currently visible (some other operations might be needed to make it visible)? If this simulation is a single haptic action, then the object should be visible from the start. If the simulation is part of a multi-action task, then the user did not perform the needed steps to make the part visible.
  - b) Object is not in a safe operable status (e.g., too hot, still running, etc.). This is often an error that can be determined by the simulation itself.
  - c) Object is too big or too heavy. (manipulation might require tools or more people). It is assumed that this is not a constraint of the haptic device (as that would be one of the previous errors), but a constraint in the real world. Again, the objective in this case is to determine what would occur in a real world situation. The haptics simulation may very well have performed exactly as intended.



## VIII. Demonstrate the Assessment & Validation Approach

*Goals: To demonstrate the assessment process on one or more identified maintenance actions and to demonstrate a notional maintenance procedure validation, depicting the salient characteristics of the validation methodology.*

For the demonstration of the assessment and validation approach, we use the example of developing the bayonet connector model simulation. There are many reasons to choose the bayonet connector model as our simulation example. First, it provides a good representative example of a haptic action to simulate, as it uses both force and torque. Second, it requires relatively low levels of force and torque, and therefore is less stressful on the existing experimental equipment. Third, the bayonet assembly has low virtual mass and is lightweight, so human strength or performance limitations are not salient. Fourth, arm access to the bayonet assembly and the concomitant reach analysis is not critical for the haptic experiments: were it necessary, the experimental setup could be repositioned or subject to physical obstructions for alternative access requirements. Finally, the bayonet connector has a general rather than specific function, so results may be more readily generalized to specific configurations. Consequently, haptic validation of the bayonet assembly provides a focused assessment of user performance on the task itself and potential analysis of user errors.

### A. Simulation Development and Setup

#### Primary Concerns

We addressed the primary concerns as follows.

- Haptic task being performed – how suitable is it for haptic simulation?

The haptic task we are performing is the simulation of a bayonet connector. The way we implemented it, this action falls into the fine motor control task category, which is rated excellent for simulation.

- Haptic devices being used – is the device suitable for the given task? Or does a new device have to be purchased?

The Phantom is arguably the most suitable haptic device for this type of simulation. Due to funding issues, it was not feasible to acquire a new haptic device.

- Development time and costs

The development time and costs were set when the contract was issued (1 January, 2001), which was before we developed this methodology for assessing actions. Thus, the time and costs required were already fixed.

We decided that the simulation of a bayonet connector would be an excellent choice for a single haptic action simulation.

### **General Concerns**

- **Purpose:** The purpose of this haptic simulation development was to begin to assess how useful haptic simulations are for aircraft maintenance training. Although the initial tasks will be individual haptic actions, the idea is that they would eventually be combined into a system that would provide a certain level of realism while providing training not available without an airplane.
- **Existing forces and torques:** The bayonet connector model is a bounded cylinder (described in section 3.5.3, CAD Geometry Development). In addition to the normal operation of the bayonet connector (the insertion and removal of the key), we did not want the key to be able to be inserted into the wrong part of the bayonet connector (the rear, the sides, etc.). Also, a button needed to be placed in the bayonet connector to simulate the "pushing" of the key as it is inserted into the connector. Friction was not a concern in this model, as one could have a nearly frictionless bayonet connector (a well-greased connector in an aircraft, for example). By nature of a bayonet connector, we needed to use both force and torque in our simulation.
- **Haptic action being performed:** We chose the bayonet connector as our action to be performed. The SMG team enumerated 6 actions that should receive priority for simulation (the bolded entries in the taxonomy table). The bayonet connector was the most interesting one of that subset. This particular action best fit into the fine motor control category, using both force and torque. Since friction was not an important component in our simulation, it did not fit in the tactile friction category. And because we decided to use the Phantom (see below), we could not place it in the multi-finger task category. From the taxonomy task evaluation effectiveness summary table, we can see that this action has an excellent potential for being simulated.
- **Haptic device being used:** Our two choices were the CyberGrasp and the Phantom. As we have both in our lab, financial constraints were not applicable. The CyberGrasp cannot apply torque, so we were able to eliminate it immediately from our consideration. This left the Phantom as our haptic device of choice. The Phantom allows for both forces and torques to be simulated, which is desirable. While the Phantom does not allow for individual finger forces, the essence of using a bayonet connector, in our case, can be done without the individual finger forces. The Phantom also has a limited range of motion (the length of the robotic arm), but this would not pose a problem, as our simulated action does not need a lot of space.

- **Host platform:** Our first choice was to use a UNIX-based environment, because of the stability, development environment, and development tools available. However, many of the libraries are only available for Microsoft Windows. Both the Phantom libraries and the libraries for the head mounted display are Windows-only releases. Thus, our decision was really made for us. We used a Microsoft Windows NT 4.0 system, running on a Pentium III, 850 Mhz.
- **Geometric model:** For our geometric model, we created the bayonet connector out of the constructive solid geometry methods described in section 3.5.3, CAD Geometry Development. Our model is not deformable, which simplified the required code. Another consideration is that the Phantom requires the developer to provide line segment intersection routines, which are very difficult for many models, but feasible for constructive solid geometry models. These routines are called 1,000 times a second by the Phantom library code, so efficient geometric models was a high priority, and CAD models were not viable.
- **Physical attribute model:** As we were simulating a single haptic action, we did not need a complicated physical attribute model.
  - **Geometry files:** We used CSG objects for our geometry files, as described in Appendix A. Since we did not have to convert them to geometrical surfaces, we did not have to increase the data set, and we were able to keep the correct level of realism by not having to approximate curved surfaces with a number of flat polygons. For the bayonet connector itself, we used a "point shell", which is a series of points whose surface approximates the surface of a bayonet connector key. This point shell can be seen on the right of figure 4, below.
  - **Assembly and structure data:** Because we used CSG objects and a point shell, we did not need any additional assembly and structural data.
  - **Physical simulations:** The force and torque simulations that we needed to simulate were handled mostly by the Phantom's library. Each of the points in the point shell would produce resistance when that point attempted to enter a solid object of the CSG shape.
  - **Graspable sites:** For this simulation, pressing the button on the Phantom's handle caused the Phantom to "grab" the cylindrical key, provided that the position of the Phantom in the simulation was very close to the key. Once grabbed, the key was an extension of the Phantom's shaft. This was all the graspable sites needed for this simulation.
- **Graphical environment:** We did not focus significantly on the graphical environment, as our primary purpose was to get the haptics aspect of it working. We wrote routines that would draw the bayonet connector on the screen, but did not feel it necessary to add realism in the form of lighting, shading, textures, etc. As described above (in section 3.5.4, Assessment of Virtual Maintenance Actions, under the Benchmarks and Comparisons Among Haptics Platforms sub-section), one would not use a single haptic

action for simulation realism analysis. Thus, if and when our system is expanded to include multiple individual haptic actions, we would add the necessary graphical realism. Another reason for the lack of high-end graphics is that there were some serious performance issues (described below), and having the computer take a lot of time to draw the scene on the screen would only exacerbate those problems (we didn't have another processor to render the graphics on). We have a head mounted display, and using it would not cause any significant decrease in performance, but since graphical realism was not required, we did not include it.

- **Development environment:** We chose to develop our code in C++. This choice was decided by a number of factors out of our control. The libraries (for the Phantom and the head mounted display) are both C/C++ libraries. Since execution time is a priority, we needed a language that could compile to efficient code, so Java was out of the picture. We chose C++ over C as a lot of the code benefited from the object oriented programming features of C++. We used Visual C++ because it is the primary C/C++ development system for Microsoft Windows, and because that's what our lab's machines had installed on it.
- **Development personnel:** There were two people who were doing the main part of the development on this project. One developer did the Phantom-side coding, which included the code that dealt with the graphics, and allowed the haptics to interact with the shapes. The other developer did the CSG part of the coding, which included all the code for interactions with the shapes formed by the CSG operations. Each of the developers thought that the other one had the harder coding task...

## B. Simulation Execution

The forces and torques of the simulation were handled by the Phantom's library, as called by the user code. The biggest issue we encountered was the haptics delay loop. The Phantom must update its haptics 1000 times a second to provide for proper haptic sensation. This means that the code that is called for each of those 1000 iterations must execute very quickly. We optimized our code so that it would execute within the required time constraints. These optimizations were achieved by decreasing the number of objects that made up the CSG shape of the bayonet connector, lowering the resolution of the bayonet key (the number of points on the point shell), and various other low-level programming optimization techniques. Our bayonet connector was a rigid object, so the simulation did not have to do the extra computation that would be necessary for a deformable object. Continued improvements in computer speed and programming experience with this example will likely result in more efficient experiment code production in the future.

An image of what the simulation looked like on the computer screen is shown in figure 4, below.

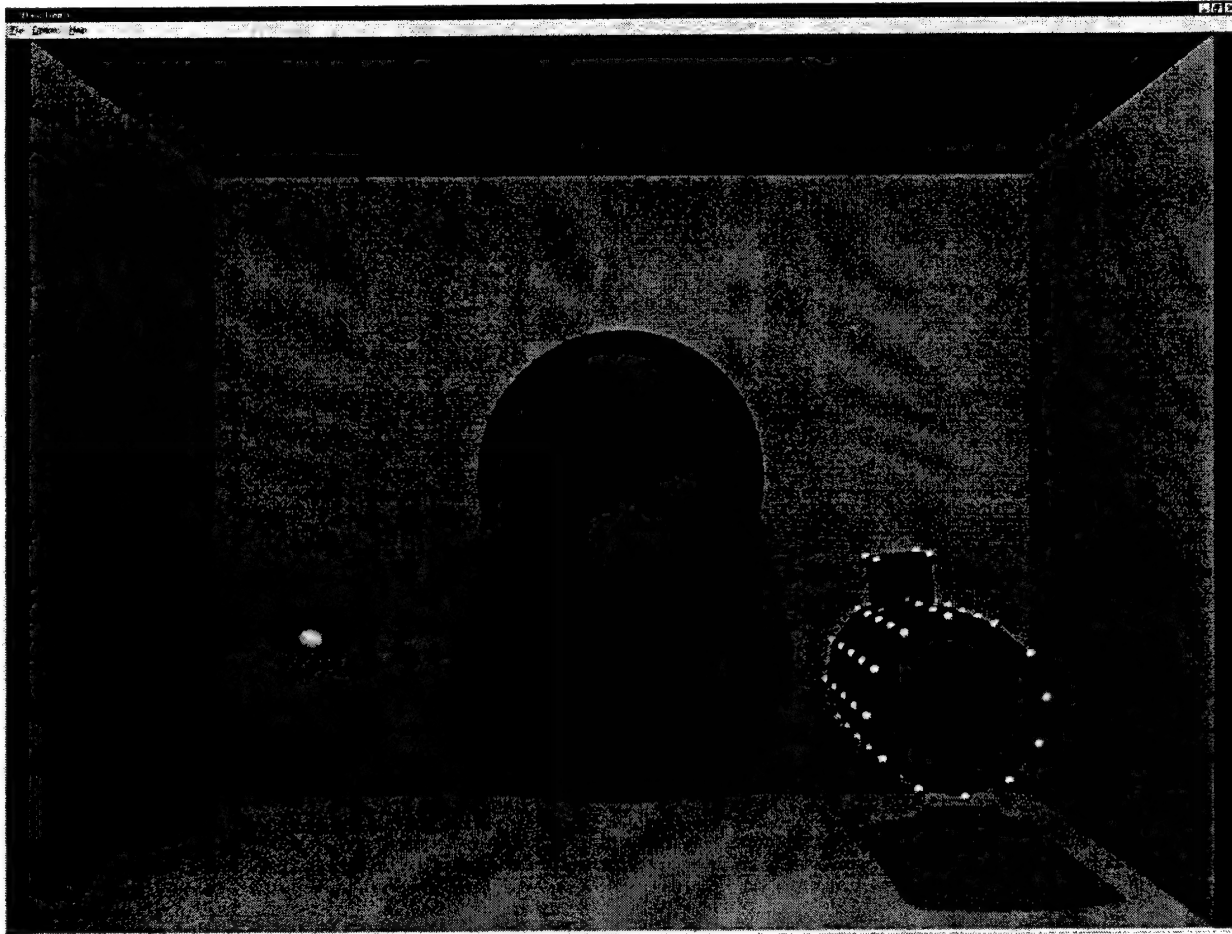


Figure 4: Bayonet connector simulation

### C. Evaluation Design

As the focus of this work effort was not on exhaustive human subject tests, and because we were simulating a single haptic action (as opposed to a series of haptic actions), we tested the simulation on only two subjects. For more rigorous tests, we would have used a variety of human subjects, with varying hand sizes.

We collected the data specified in the subjective analysis from the previous section, and observed which parts of the objective analysis were not met.

### D. Analysis

As noted in the previous section, an analysis on a single haptic action will not yield a particularly interesting analysis unless it is a very uncommon action (such as using a surgeon's scalpel). Using a bayonet connector is not an uncommon action. We have still provided the analysis below, to illustrate how to perform it.

## Subjective Analysis

After the subjects used the simulation, they answered the following questions. We used two human subjects. The numerical score that follows is the average of the scores of the two subjects that tested the simulation.

- How realistic did the graphics of the simulation seem? (1 = totally unrealistic, 5 = completely realistic)

Rating: 4. Although the graphics did not look like the real world, the graphics that were provided did properly display the bayonet connector. The graphical display is shown in figure 4, above.

- How realistic did the haptics of the simulation seem? (1 = totally unrealistic, 5 = completely realistic)

Rating: 3. Due to limitations with the Phantom (it cannot handle solid object interactions easily), and the Phantom's buggy libraries, the haptic realism would vary from a 2 to a 4, depending on how it was working that day.

- How realistic did the haptics of the simulation seem compared to the real world? (1 = totally unrealistic, 5 = completely realistic)

Rating: 2. Because of the buggy libraries, and the limitations of the Phantom (the grasp of the shaft), the haptics realism was decent.

- How realistic did the simulation feel compared to the real-world situation? (Only valid if they have done the real world action) (1 = totally unrealistic, 5 = completely realistic)

Rating: 2. Many bayonet connectors will have the user holding a cylindrical object (BNC connector, medicine bottle). Thus, the realism was fair.

- How comfortable would you feel performing the task in the real world? (1 = not comfortable at all, 5 = completely comfortable)

Rating: 3. Note that this simulation was a single haptic action, and a very common action at that. The subjects felt that they did not learn much more about the action, as they felt

they already knew how to use a bayonet connector. Thus, the simulation did not increase their level of comfort.

- How much did you feel you learned about this task through the simulation (1 = nothing, 5 = as much as one could learn about the task)

Rating: 1. Same reason as for the previous question: this simulation was a single haptic action, and a very common action at that. The subjects felt that they did not learn much more about the action, as they felt they already knew how to use a bayonet connector. Thus, the simulation did not teach them much about the task.

- How realistic did the grasp simulation seem? (1 = totally unrealistic, 5 = completely realistic)

Rating: 4. The Phantom requires the user to grasp its shaft. In this simulation, the key of the bayonet connector was also a cylinder, so this allowed for a lot of realism.

- How realistic did the magnitude of the forces and torques seem? (1 = totally unrealistic, 5 = completely realistic)

Rating: 3. The bayonet connector simulation did not require a lot of force or torque. This was fairly realistic, as many real world bayonet connectors also do not require a lot of force or torque.

- How helpful did you feel this simulation was? (1 = not helpful at all, 5 = very helpful)

Rating: 1. Same reason as for the "how much learned" question: this simulation was a single haptic action, and a very common action at that. The subjects felt that they did not learn much more about the action, as they felt they already knew how to use a bayonet connector. Thus, the simulation was not that helpful.

## **Objective Analysis**

The objective analysis was performed by observing the subject interacting with the simulation, and recording the results shown below.

1. Which user errors did the subject encounter?

The users encountered the following errors. These were user errors, and were corrected in future runs of the simulation. Note that the subjects purposely tried to encounter some of these errors in some of the simulation runs.

- a) User exerts too much force or too little force
- b) User exerts too much torque or too little torque
- c) User exerts force in the wrong direction
- d) User exerts torque in the wrong direction or around a wrong axis
- e) User exerts force too early or too late
- f) User exerts torque too early or too late

## 2. Device Limitations:

- a) Was the maximum force or torque that the haptic device could safely provide insufficient to model the physical system?

The Phantom was able to apply sufficient force and torque.

- b) Was the maximum movement allowed by the device is insufficient to model the required action?

No. The Phantom allowed sufficient movement for the simulation.

- c) Was the haptic device not mobile, thus preventing proper reproduction of the physical system?

Although the Phantom was not mobile, that was not a problem, as the range of motion that the stationary Phantom required was sufficient for the simulation.

- d) Does the grasp require friction or tactile feedback not provided by the haptic device?

Our simulation did not need to use friction, as the action could be simulated without it.



3. Insufficient access: The task required manipulating the hand/arm in such a way that environmental constraints (collisions) between the arm and other objects would have prevented the motion.

The only objects in the environment were the bayonet connector and the key for the bayonet connector. Thus, there were no other environmental constraints.

4. Insufficient function: Would the task have caused a response (e.g., a hazard, vibration, shock, temperature change, fluid release, etc.) in the physical system that could not be represented via haptic feedback?

No. As this was a single haptic action, this did not apply.

5. Ability Limitations: User could not complete the operations because:

- a) Was the object to operate on not currently visible (some other operations might be needed to make it visible)?

As this was a single haptic action, the necessary objects were always visible.

- b) Object is not in a safe operable status (e.g., too hot, still running, etc.).

As this was a single haptic action, this limitation did not apply.

- c) Object is too big or too heavy (manipulation might require tools or more people).

As this was a single haptic action, this limitation did not apply.

### **Results for this Maintenance Procedure**

From the simulation analysis results, we can tell that there is a lot of room for improvement. This improvement consists of many possibilities, including multiple haptic actions in a single simulation, better design of (and use of) the Phantom libraries, and more graphical detail. These improvements are part of the focus for the continuation grant of this work order.

## IX. Conclusion

The goals of this study are threefold: 1) development of a maintenance action taxonomy, 2) assessment of virtual task validation using haptics, and 3) development of a virtual validation evaluation methodology.

A task taxonomy was developed to provide a framework for further maintenance action analysis. The taxonomy facilitates understanding specific haptic simulation benefits, evaluation designs, errors, and limitations. In general, haptic feedback for maintenance tasks requiring strength or constrained body configurations is strongly limited by present equipment capabilities. Manipulations requiring low force precision hand movements are somewhat better served by haptic feedback. Human factors experiments should be undertaken to test user abilities with haptic feedback against both non-haptic and actual physical manipulation. Alternative simulation approaches were investigated that might be used in combination with, substitution for, or augmentation of haptics to meet instruction validation goals. Our recommendations may be used to guide or further focus efforts in the SMG program.

We examined several cases in our assessment of haptics validation:

1. Interactive user task attempts and analysis
  - a. Using visual and haptic feedback
  - b. Using visual feedback only
2. Non-interactive task attempts and analysis

Cases 1a and 1b provide alternatives to haptics and must be explored even if only as experimental controls.

Any claims about the veracity or usefulness of haptics simulations for maintenance actions ought to be measured against similar tasks executed without haptic feedback. A baseline non-haptic simulation (case 1b) would involve a visual interface but use manual interaction devices without haptic feedback. In general, such devices would be standard computer input devices such as a mouse, keyboard, joystick, or trackball. Note that a CyberGlove is not a "standard" input device (meaning it is not something the average computer user will own). Manipulating these devices would cause CAD objects to move interactively (real-time) on screen. Purely visual feedback on task progress would be displayed. Such an arrangement would typically include camera controls so that the user view could be readily changed to any suitable position and visual feedback for at least collision detection. In addition, visual feedback might be available to monitor forces and torques needed relative to human maintainer capabilities. In an ideal situation, such limitations would actually be used as constraints on the allowable movements executed by the user. At present, we know of no software tools that enforce such constraints during interactive manipulation. This is a possible area for future algorithm research and development.

In case 2, "pure" computation would be used to establish task validity. What this means is that a computer simulation would need to establish that a given maintenance task could or could not be performed. There are four possible outcome cases:

- A. The task is physically possible and humanly possible by any "typical" aircraft maintainer.
- B. The task is physically possible but unreasonable to expect from a "typical" aircraft maintainer (e.g., insufficient strength).
- C. The task is physically impossible due to human limitations (of any maintainer).
- D. The task is physically impossible due to physical limitations (part is just inaccessible or not extractable).

Clearly this problem is complicated by the need to characterize aircraft maintainers and their statistical capabilities with respect to anthropometry, dexterity, strength, and skill. (Our haptic study does not overtly address some of these variables though they are critically important to assess situations A, B, and C.) Existing human form models do cover some of these critical variables. Independent of how well or not such human form models parameterize this space, it is crucial to note that any interactive simulation based on a real user will represent solutions based on a sample set of one, and thus will not provide any more broad parameterization than existing human form models: a single user is not a statistically useful data point for task validation (with or without haptics!) except possibly for cases C and D. Human factors experimenters will typically use several subjects to assess task validity for cases A and B. Such multiple subject tests are clearly possible and desirable for statistical purposes.

Increasing the number of subjects that attempt to validate a task increases both cost and set-up time. Human form models address this accommodation problem directly by allowing the user to manipulate or test task validity with multiple models computed or selected from known anthropometric populations (such as aircraft maintainers). It is not known at this time whether virtual interactive simulations with haptic feedback will be more or less costly than non-haptic simulations or tests on actual physical devices (mock-ups or the actual aircraft). Such comparisons should be undertaken in future studies.

So returning to the outcomes A-D above, task validation requires establishing which of A-D are false. Can "pure" computation (case 2) play a role? We believe the answer is affirmative, but it depends on developing new software approaches to human modeling. The major issues are:

- Reach algorithms must find access paths in confined spaces or determine that no solution can be found.
- These algorithms must take into account body size, articulation, joint limits, soft tissue deformations, tool handling, and clothing restraints and thickness in order to make accurate reachability assessments.
- These algorithms must respect human torque and strength limitations.

- These algorithms must also address multi-point bracing and contacts as leverage to strategize and complete tasks.

Although no existing reach algorithm meets all these goals, it remains a desirable research objective. We believe that such an algorithm is possible but it will take dedicated development. Funding opportunities should consider this option in parallel to interactive and haptic feedback methods, since these other operational approaches (including haptics) cannot presently satisfy the overall validation goal entirely either.

The development of a virtual validation evaluation methodology will necessarily combine simulation, interaction, and human skills to inform designers of maintainability problems and design flaws. The mitigation of discovered flaws may be partially automated, but human designers will play the major role. Integrating necessary data from product and assembly databases, determining and using engineering parameters, creating or obtaining appropriately detailed CAD models, and coding efficient and effective kinematic, dynamic, haptic, and human performance simulations will both challenge and motivate this validation vision in the future.

## X. Appendix A

### A. Modeling the Bayonet Connector

We first modeled a large bounded cylinder and a small bounded cylinder by intersecting infinite cylinders with a box. The difference operator yielded a tube. The inner cylinder, shown in orange in figure 5, is being subtracted from the outer cylinder.

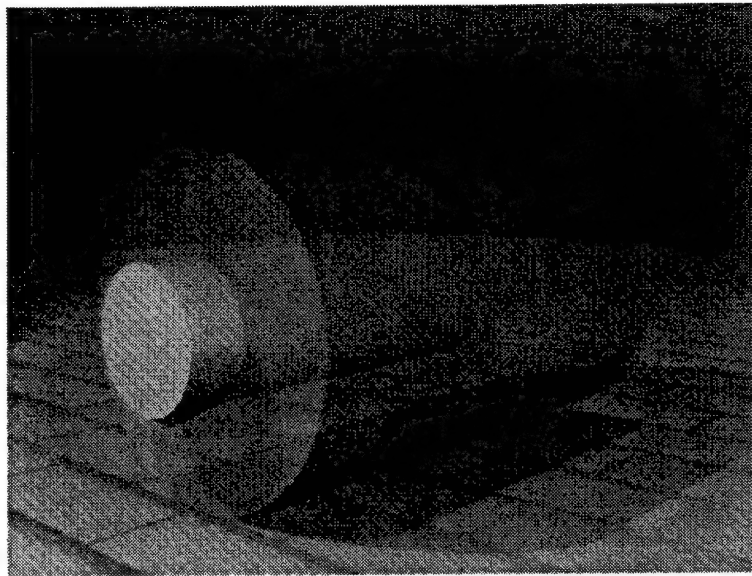


Figure 5: Bayonet connector: the tube

The *insertion groove* is the long groove that the notch on the key is inserted into when you first put the key into the bayonet connector. The wedge that is subtracted from the tube to form the insertion groove is shown in orange in figure 6. Note that we used a box instead of a wedge, but the concepts are the same.



Figure 6: Bayonet connector: the insertion groove

The *rotation groove* is the curved groove that the notch moves within as the key is rotated within the bayonet connector. The partial cylinder that is subtracted from the tube to form the rotation groove is shown in orange in figure 7. Intersecting an infinite cylinder with two separate boxes forms that cylinder.

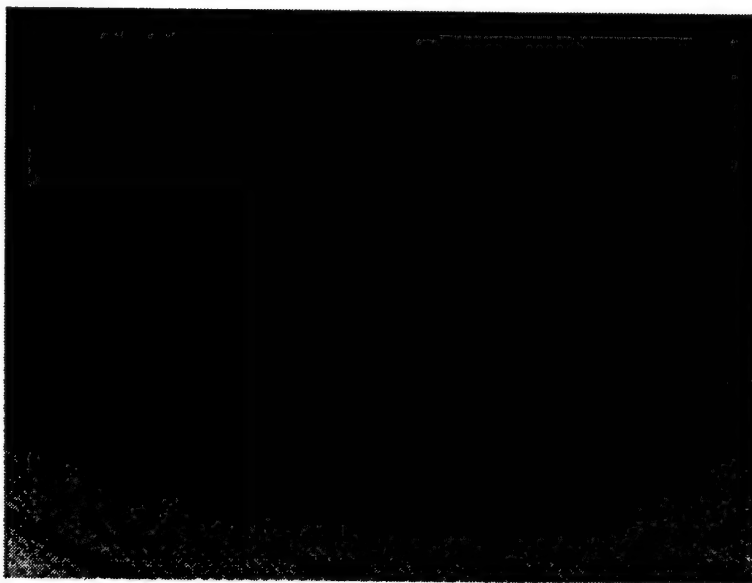


Figure 7: Bayonet connector: the rotation groove

The *lock groove* is the groove that the notch moves within as the key moves outward and locks into place. The wedge that is subtracted from the tube to form the lock groove is shown in orange in figure 8. Note that we used a box instead of a wedge, but the concepts are the same.

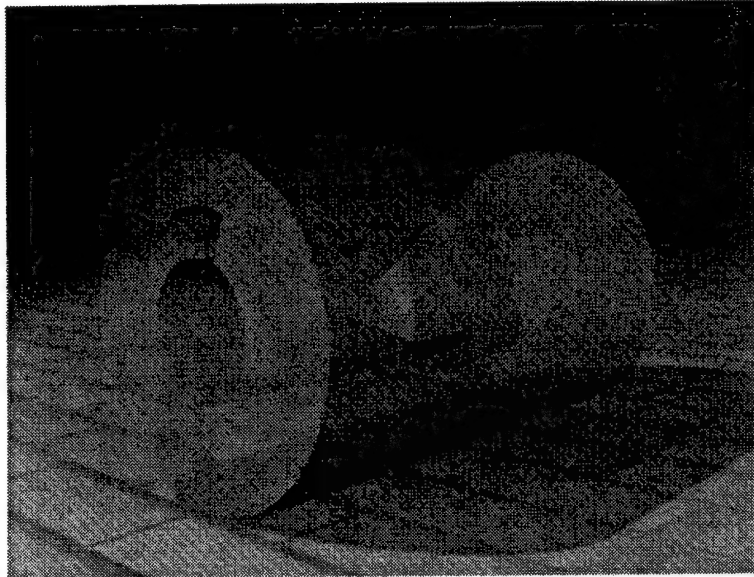


Figure 8: Bayonet connector: the lock groove

The entire model is shown in figure 9. It is difficult to see the details of the inside grooves, so we also show a wire frame version of the completed model in figure 10.

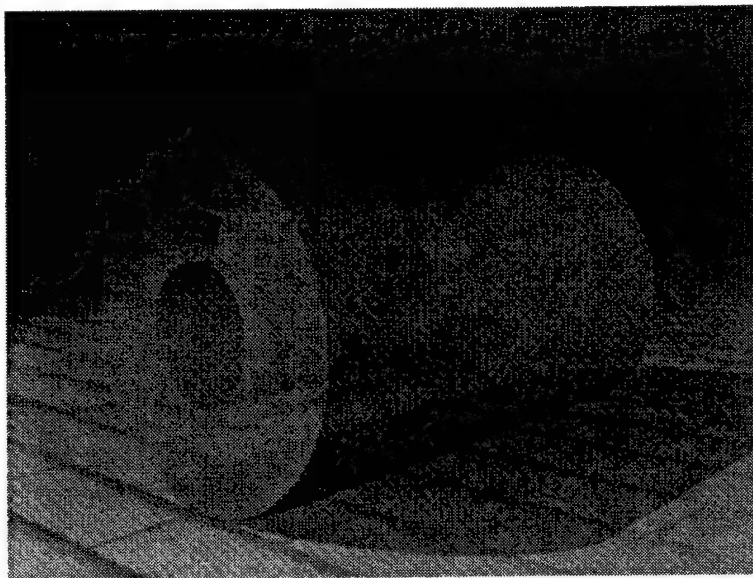


Figure 9: Bayonet connector: completed transparency view

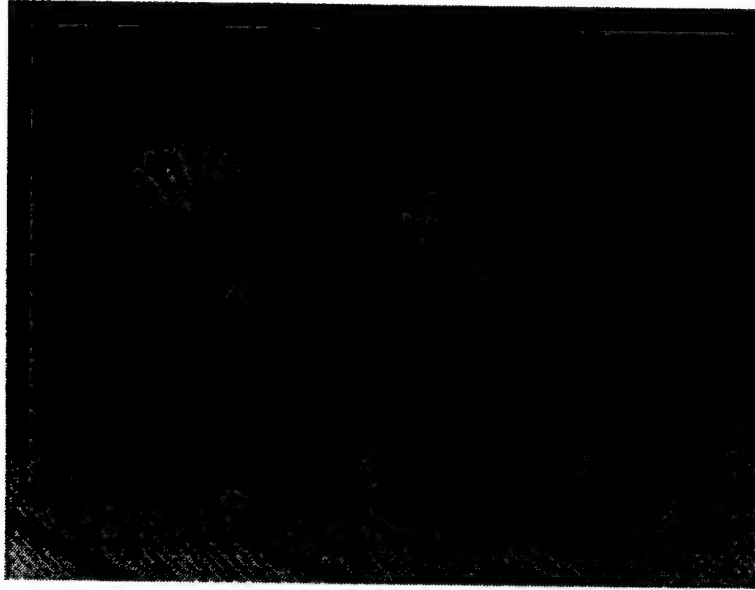


Figure 10: Bayonet connector: completed wireframe view

Lastly, the key, with the notch that fits into the grooves described, is shown in figure 11.

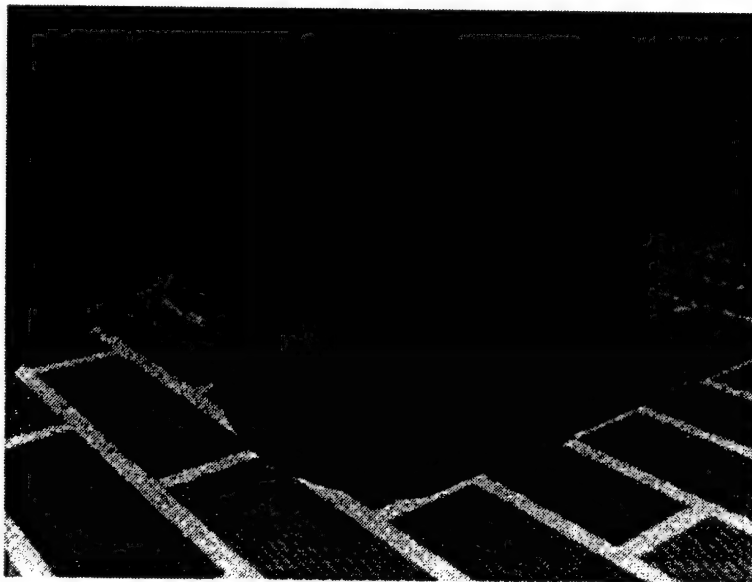


Figure 11: Key for the bayonet connector

Note that the model shown above and on the web page is what the haptics environment would see, and is not necessarily what is displayed on the screen. Thus, to increase realism, we can display real-life controls instead of the model. These models would most likely consist of the various geometry models that we have already received. This allows for the haptics interactions to work as desired, as well as for visual realism.



## XI. References

AKH01

Richard J. Adams, Daniel Klowden, and Blake Hannaford. Virtual Training for a Manual Assembly Task. *Haptics-e* 2:2, October, 2001, <http://www.haptics-e.org>

AIL85

M. A. Arbib, T. Iberall, and D. M. Lyons. Coordinated control programs for movements of the hand. In S. W. Goodwin and I. Darian-Smith, editors, *Hand Function and the Neocortex*, pages 111-129, Berlin, 1985. Springer-Verlag.

AS96

Ricardo S. Avila and Lisa M. Sobierajski. A haptic interaction method for volume visualization. In *Proceedings of the Conference on Visualization*, page 197, 1996.

BBE+00

Norman Badler, Rama Bindiganavale, Charles Erignac, Patrick Vincent. Technology Assessment for Automated Technical Order Validation. November 2000, Final Report to Air Force Research Laboratory, Human Effectiveness Directorate (AFRL/HE), Logistics Sustainment Branch (HESS).

BC94

Grigore Burdea and Philippe Coiffet. *Virtual Reality Technology*. John Wiley & Sons, New York, 1994.

Bur96

Grigore C. Burdea. *Force and Touch Feedback for Virtual Reality*. John Wiley & Sons, New York, 1996.

Cha01

Don B. Chaffin. *Digital Human Modeling for Vehicle and Workplace Design*. Society of Automotive Engineers, Warrendale, PA, 2001.

CPK+98

Chetz Colwell, Helen Petrie, Diana Kornbrot, Andrew Hardwick, and Stephen Furner. Haptic virtual reality for blind computer users. In *Third Annual ACM Conference on Assistive Technologies*, pages 92-99, 1998.

Cut89

Mark R. Cutkosky. On grasp choice, grasp models, and the design of hands for manufacturing tasks. *IEEE Journal of Robotics and Automation*, 5(3):269-279, June 1989.

Cybf01

Immersion Corporation - 3D Technologies CyberForce at <http://www.immersion.com/products/3d/interaction/cyberforce.shtml>.

Cygl01

Immersion Corporation - 3D Technologies CyberGlove at <http://www.immersion.com/products/3d/interaction/cyberglove.shtml>.

Gygr01

Immersion Corporation - 3D Technologies CyberGrasp at <http://www.immersion.com/products/3d/interaction/cybergasp.shtml>.

DP00

Derek DiFilippo and Dinesh K. Pai. The AHI: an audio and haptic interface for contact interactions. In *Proceedings of the 13th Annual Symposium on User Interface Software and Technology (UIST-00)*, pages 149-158, N.Y., November 5-8 2000. ACM Press.

DQKES99

Frank Dachille IX, Hong Qin, Arie Kaufman, and Jihad El-Sana. Haptic sculpting of dynamic surfaces (color plate S. 227). In Stephen N. Spencer, editor, *Proceedings of the Conference on the 1999 Symposium on interactive 3D Graphics*, pages 103-110, New York, April 26-28 1999. ACM Press.

GME+00

Arthur Gregory, Ajith Mascarenhas, Stephen Ehmann, Ming Lin, and Dinesh Manocha. Six degree-of-freedom haptic display of polygonal models. In T. Ertl, B. Hamann, and A. Varshney, editors, *Proceedings Visualization 2000*, pages 139-146. IEEE Computer Society technical Committee on Computer Graphics, 2000.

Ibe87

T. Iberall. The nature of human prehension: Three dextrous hands in one. In *Proceedings 1987 IEEE International Conference on Robotics and Automation*, pages 396-401, Raleigh, NC, 1987.

Isd01

Jerry Isdale. Haptics. VRNews 10:3, March-April 2001.  
<http://www.vrnews.com/issuearchive/vrn1003/vrn1003tech.html>

JBm+00

David Jack, Rares Boian, Alma Merians, Sergei V. Adamovich, Marilyn Tremaine, Michael Recce, Grigore C. Burdea, and Howard Poizner. A virtual reality-based exercise program for stroke rehabilitation. In *Fourth Annual ACM Conference on Assistive Technologies*, pages 56-63. ACM, 2000.

JJW+99

Sankar Jayaram, Uma Jayaram, Yong Wang, Hrishikesh Tirumali, Kevin Lyons, Peter Hart. VADE: A Virtual Assembly Design Environment. *IEEE Computer Graphics & Applications* 19:6, pp. 44-50, 1999.

Kur97

Martin Kurze. Rendering drawings for interactive haptic perception. In *Proceedings of ACM CHI 97 Conference on Human Factors in Computing Systems*, volume 1 of *PAPERS: Speech, Haptic, & Multimodal Input*, pages 423-430, 1997.

LLPN00

Dale A. Lawrence, Christopher D. Lee, Lucy Y. Pao, and Roman Y. Novoselov. Shock and vortex visualization using a combined visual/haptic interface. In T. Ertl, B. Hamann, and A. Varshney, editors, *Proceedings Visualization 2000*, pages 131-137. IEEE Computer Society Technical Committee on Computer Graphics, 2000.

MD97

Stefan Munch and Rudiger Dillmann. Haptic output in multimodal user interfaces. In *Proceedings of the 1997 International Conference on Intelligent User Interfaces, I/O Support/Spatial Awareness*, pages 105-112, 1997.

MPT99

William A. McNeely, Kevin D. Puterbaugh, and James J. Troy. Six degrees-of-freedom haptic rendering using voxel sampling. In Alyn Rockwood, editor, *SIGGRAPH 1999*, Annual Conference Series, pages 401-408, Los Angeles, 1999. ACM, Addison Wesley Longman.

MZ98

Timothy Miller and Robert Zeleznik. An insidious haptic invasion: Adding force feedback to the X desktop. In *Proceedings of the ACM Symposium on User Interface Software and Technology, More than a Mouse*, pages 59-64, 1998.

MZ99

Timothy Miller and Robert C. Zeleznik. The design of 3D haptic widgets. In Stephen N. Spencer, editor, *Proceedings of the Conference on the 1999 Symposium on interactive 3D Graphics*, pages 97-102, New York, April 26-28 1999. ACM Press.

Nap56

J. R. Napier. The prehensile movements of the human hand. In *Journal of Bone and Joint Surgery*, volume 38B(4), pages 902-913, 1956.

NMK96

Haruo Noma, Tsutomu Miyasato, and Fumio Kishino. A palmtop display for dextrous manipulation with haptic sensation. In Michael J. Tauber, Victoria Bellotti, Robin Jeffries, Jock D. Mackinlay, and Jakob Nielsen, editors, *Proceedings of the Conference on Human Factors in Computing Systems: Common Ground*, pages 126-133, New York, April 13-18 1996. ACM Press.

Phan01

SensAble Technologies: Haptics Research at <http://www.sensable.com/haptics/products/phantom.html>.

Ram96

Christophe Ramstein. Combining haptic and Braille technologies: Design issues and pilot study. In *Second Annual ACM Conference on Assistive Technologies, Vision Impairments - I*, pages 37-44, 1996.

RH94

Christophe Ramstein and Vincent Hayward. The pantograph: A large workspace haptic device for multi-modal human-computer interaction. In *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems*, volume 2 of *INTERACTIVE EXPERIENCE*, pages 57-58, 1994.

RKK97

Diego C. Ruspini, Krasimir Kolarov, and Oussama Khatib. The haptic display of complex graphical environments. In Turner Whitted, editor, *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series, pages 345-352. ACM SIGGRAPH, Addison Wesley, August 1997. ISBN 0-89791-896-7.

RYB+00

Rameshsharma Ramloll, Wai Yu, Stephen Brewster, Beate Riedel, Mike Burton, and Gisela Dimigen. Constructing sonified haptic line graphs for the blind student: First steps. In *Fourth Annual ACM Conference on Assistive Technologies*, pages 17-25. ACM, 2000.

Sc119

G. Sclesinger. Der mechanische aufbau der kunstlichen glieder (the mechanical building of artificial limbs). In M. Borchardt, editor, *Ersatzglieder und Arbeitshilfen für Kriegsbeschädigte und Unfallverletzte (Prosthetic and Occupational Help for War Victims and Accident Victims)*, pages 321-699. Springer, 1919.

TJC97

Thomas V. Thompson, II, David E. Johnson, and Elaine Cohen. Direct haptic rendering of sculptured models. In *Proceedings of the Symposium on Interactive 3D Graphics*, pages 167-176, New York, April 27-30 1997. ACM Press.

TS55

C. L. Taylor and R. J. Schwarz. The anatomy and mechanics of the human hand. In *Artificial Limbs*, 2, pages 22-35, 1955.

VI88

S. T. Venkataraman and T. Iberall. *Dextrous Robot Hands*. Springer-Verlag, New York, 1988.

WM00

Yanqing Wang and Christine L. MacKenzie. The role of contextual haptic and visual constraints on object manipulation in virtual environments. In *Proceedings of ACM CHI 2000 Conference on Human Factors in Computing Systems*, volume 1 of *3D Input*, pages 532-539, 2000.

WZM+01

Roger Webster, D. Zimmerman, B. Mohler, M. Melkonian, R. Haluck, A Prototype Haptic Suturing Simulator. In Proceedings of the Medicine Meets Virtual Reality Conference, (MMVR '2001), Newport Beach, California, Aligned Management Association and the University of California at Irvine, IOS Press, January 24-27, 2001, pp. 567-569.

WTT92

Wesley Woodson, Barry Tillman, and Peggy Tillman. *Human Factors Design Handbook*. McGraw-Hill, New York, 1992.